NASA TECHNICAL MEMORANDUM

NASA TM X-2375

NASA TM X-2375

# COMPUTER PROGRAM FOR THE TRANSIENT RESPONSE OF ABLATING AXISYMMETRIC BODIES INCLUDING THE EFFECTS OF SHAPE CHANGE

*by Lona M. Howser and Stephen S. Tompkins*

*Langley Research Center*
*Hampton, Va. 23365*

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • OCTOBER 1971

| 1. Report No. | 2. Government Accession No. | 3. Recipient's Catalog No. |
|---|---|---|
| NASA TM X-2375 | | |

4. Title and Subtitle
COMPUTER PROGRAM FOR THE TRANSIENT RESPONSE OF ABLATING AXISYMMETRIC BODIES INCLUDING THE EFFECTS OF SHAPE CHANGE

5. Report Date
October 1971

6. Performing Organization Code

7. Author(s)
Lona M. Howser and Stephen S. Tompkins

8. Performing Organization Report No.
L-7900

10. Work Unit No.
136-13-05-04

9. Performing Organization Name and Address
NASA Langley Research Center
Hampton, Va. 23365

11. Contract or Grant No.

13. Type of Report and Period Covered
Technical Memorandum

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, D.C. 20546

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract
A computer program to analyze the transient response of an ablating axisymmetric body including the effects of shape change is presented in detail. The program, its subroutines, and their variables are listed and defined. The computer input and output, in printed and plotted form, for three sample problems are presented to aid the user in setting up and running a problem with the program. The governing differential equation, the boundary conditions for the analysis on which the computer program is based, and the method of solution of the resulting finite-difference equations are discussed.

17. Key Words (Suggested by Author(s))
Ablation
Heat transfer
Computer program

18. Distribution Statement
Unclassified – Unlimited

| 19. Security Classif. (of this report) | 20. Security Classif. (of this page) | 21. No. of Pages | 22. Price* |
|---|---|---|---|
| Unclassified | Unclassified | 81 | $3.00 |

*For sale by the National Technical Information Service, Springfield, Virginia 22151

CONTENTS

Page

# COMPUTER PROGRAM FOR THE TRANSIENT RESPONSE OF ABLATING AXISYMMETRIC BODIES INCLUDING THE EFFECTS OF SHAPE CHANGE

By Lona M. Howser and Stephen S. Tompkins
Langley Research Center

## SUMMARY

A computer program to analyze the transient response of an ablating axisymmetric body including the effects of shape change is presented in detail. The program, its subroutines, and their variables are listed and defined. The computer input and output, in printed and plotted form, for three sample problems are presented to aid the user in setting up and running a problem with the program. The governing differential equation, the boundary conditions for the analysis on which the computer program is based, and the method of solution of the resulting finite-difference equations are discussed.

## INTRODUCTION

A numerical analysis of the transient response of an ablating axisymmetric body including the effects of shape change is presented in reference 1. The present paper briefly describes the analysis in reference 1 and presents in detail the associated computer program (program D2430) developed at the Langley Research Center. This paper also provides the user with an operating manual for the program.

Some of the features of the analysis and the associated program are (1) the ablation material is considered to be orthotropic with temperature-dependent thermal properties; (2) the thermal response of the entire body is considered simultaneously; (3) the heat transfer and pressure distribution over the body are adjusted to the new geometry as ablation occurs; (4) the governing equations and several boundary-condition options are formulated in terms of generalized orthogonal coordinates for fixed points in a moving coordinate system; (5) the finite-difference equations are solved implicitly; and (6) other instantaneous body shapes can be displayed with a plotting routine.

The computer program is written in the FORTRAN IV language for the Control Data 6000 series digital computer with the SCOPE 3.0 operating system. The equations have been programed so that either the International System of Units or the U.S. Customary Units may be used.

## SYMBOLS

$A = \frac{1}{x_b} \frac{\partial \delta}{\partial \xi}$

$A_c$ constant in oxidation equation corresponding to specific reaction rate

$A_j, B_j, C_j, D_j$ coefficients in equations (6)

$A_s$ constant in sublimation equation

$B_c$ constant in exponential of oxidation equation corresponding to activation energy

$B_s$ constant in exponential of sublimation equation

$C$ oxygen concentration by mass

$c_p$ specific heat

$H$ total enthalpy

$\Delta H_c$ heat of combustion

$\Delta H_s$ heat of sublimation

$h_1, h_2, h_3$ coordinate scale factors (eqs. (2))

$K$ reaction-rate constant for oxidation (eq. (10))

$k$ thermal conductivity

$L$ number of stations in x-direction

$M$ molecular weight of gas

$M_{O_2}$ molecular weight of oxygen

$m,n$ integers

| | |
|---|---|
| $\dot{m}$ | mass loss rate |
| $\dot{m}_c$ | mass loss rate due to combustion |
| $\dot{m}_s$ | mass loss rate due to sublimation |
| $p$ | exponent of pressure in sublimation equation (eq. (12)) |
| $p_w$ | wall pressure |
| $q_C$ | convective heating rate to nonablating cold wall |
| $q_{C,net}$ | hot-wall convective heating rate corrected for transpiration (eq. (9)) |
| $q_r$ | radiant heating rate |
| $R$ | radius of curvature of base curve |
| $R_{cyl}$ | cylindrical radius from axis of symmetry to base curve |
| $R_{stag}$ | stagnation-point radius of curvature |
| $r$ | exponent of radius in sublimation equation (eq. (12)); spherical coordinate |
| $S$ | number of stations in y-direction |
| $T$ | temperature |
| $t$ | thickness of heat sink |
| $w,z$ | Cartesian coordinates (sketch 2) |
| $x,y$ | curvilinear coordinates (sketch 1) |
| $x_b$ | length of base curve |
| $\alpha$ | absorptance |
| $\alpha_c$ | weighting factor for transpiration effectiveness of mass loss due to combustion |

$\alpha_s$ weighting factor for transpiration effectiveness of mass loss due to sublimation

$\beta$ either 0 or 1 depending on whether transpiration or ablation theory is used

$\delta$ material thickness

$\epsilon$ emittance

$\xi,\eta$ dimensionless curvilinear coordinates (eqs. (3))

$\theta$ angle between $R$ and $R_{cyl}$ (sketch 1); spherical coordinate

$\lambda$ mass of char removed per unit mass of oxygen

$\rho$ density of material

$\sigma$ Stefan-Boltzmann constant

$\tau$ time

$\psi$ angle between axis of symmetry and normal to surface (sketch 1)

Subscripts:

e edge of boundary layer

w wall condition

x,y coordinates

$\xi,\eta$ dimensionless coordinates

Superscripts:

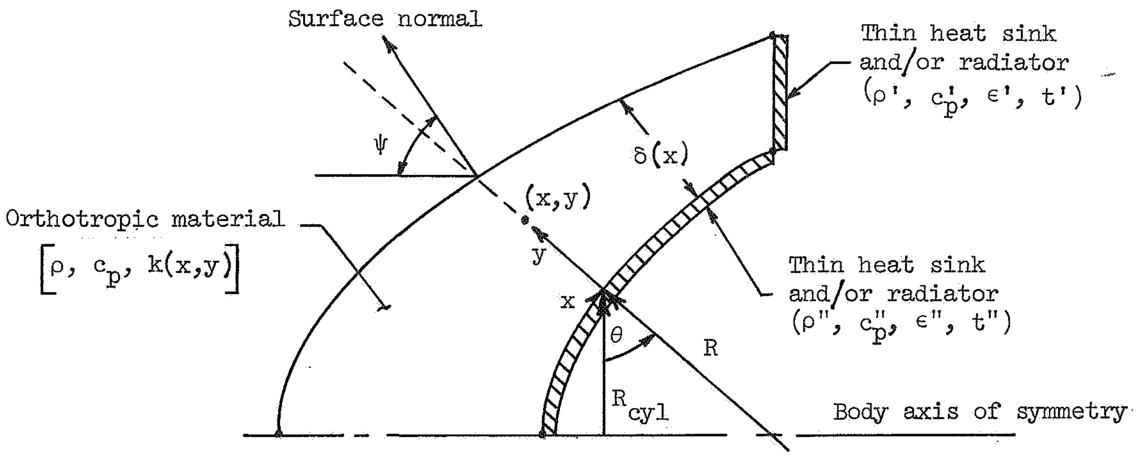
$'$ condition along $x = L$

$''$ condition along $y = 0$

## DESCRIPTION OF MODEL

### Physical Model

The analysis considers an axisymmetric ablating body exposed to aerodynamic heating; this body is composed of a single orthotropic material of varying thickness with temperature-dependent thermal properties. (See sketch 1.) The back surface of the body may be considered as a thin heat sink and/or radiator. Two coordinate systems are used to study the thermal and ablative response of the body. One is a curvilinear coordinate system, with x,y coordinates (sketch 1), which is used to determine internal temperature distributions. A stationary base curve located at the back surface of the body establishes the x-axis.




Sketch 1

The second coordinate system (sketch 2) is used to define the exterior geometry of the body which changes with time as a result of ablation. This coordinate system, with w,z coordinates, is a Cartesian system with the origin fixed at the original stagnation point of the body. All the geometric parameters needed to compute changes in the stagnation heating rates and the heating-rate and pressure distributions over the surface are defined in this system.

The governing time-dependent heat-conduction equation with variable coefficients for an axisymmetric body is, in fixed coordinates,

$$\frac{1}{h_1 h_2 h_3}\left[\frac{\partial}{\partial x}\left(\frac{h_2 h_3}{h_1} k_x \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{h_1 h_3}{h_2} k_y \frac{\partial T}{\partial y}\right)\right] = \rho c_p \frac{\partial T}{\partial \tau} \qquad (1)$$

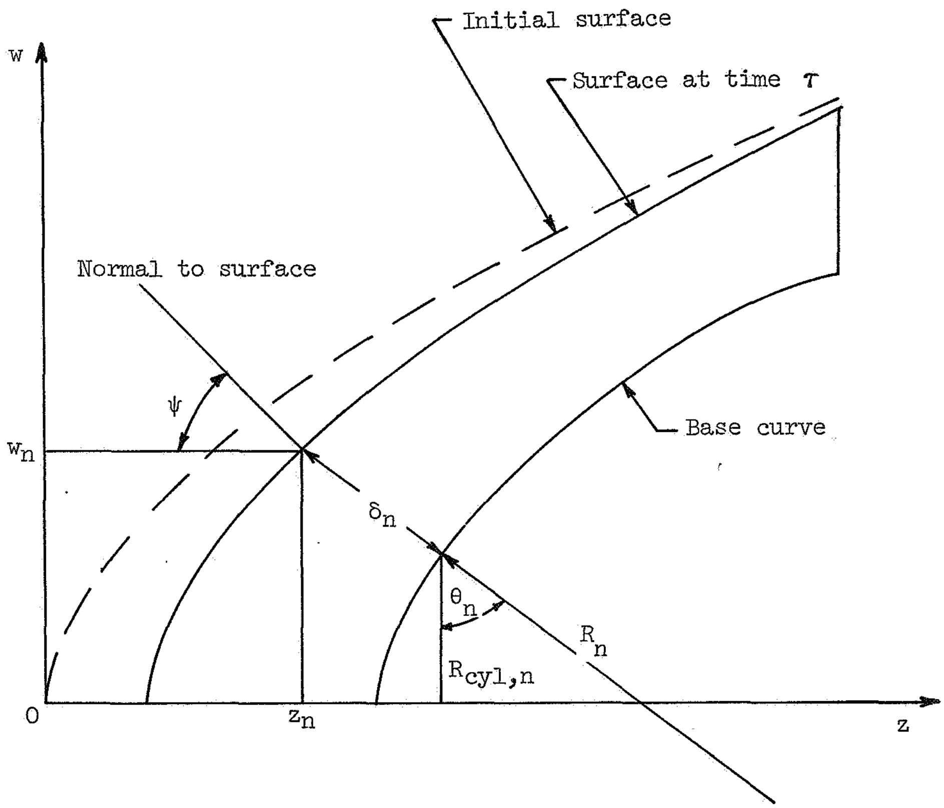
where the coordinate scale factors are

$$h_1 = 1 + \frac{y}{R} \tag{2a}$$

$$h_2 = 1 \tag{2b}$$

$$h_3 = R_{cyl} + y \cos \theta \tag{2c}$$

The transient temperature response of an ablating axisymmetric body is obtained from the solution of equation (1) with the appropriate boundary conditions, which are presented in reference 1. The method of solution is discussed in the following section.




Sketch 2

## Mathematical Model and Solution

The finite-difference method was used to obtain the solution to equation (1). However, if equation (1) were expressed in finite-difference form, it would describe the temperature variation at fixed stations in a fixed coordinate system. To maintain a fixed number of stations in a layer which changes thickness with time, it is necessary to change

the location of the stations and to interpolate to determine the temperatures at the new location after each time step. This procedure is time consuming and introduces a small error in each step of the calculation. This difficulty can be eliminated by transforming the equation to a coordinate system in which the stations remain fixed and the coordinates themselves move to accommodate changes in the surface location.

This transformation can be made by introducing a moving coordinate system $\xi,\eta$, where

$$\xi = \frac{x}{x_b} \quad \text{and} \quad \eta = \frac{y}{\delta} \tag{3}$$

In this system, the outer surface remains fixed at $\eta = 1$ and all other stations remain at fixed values of $\eta$.

The governing time-dependent heat-conduction equation (eq. (1)) in this transformed moving coordinate system is (eq. (9) in ref. 1):

$$\frac{1}{h_1 h_3}\left[\frac{1}{\delta^2}\frac{\partial}{\partial\eta}\left(h_1 h_3 k_\eta \frac{\partial T}{\partial\eta}\right) + \frac{1}{x_b{}^2}\frac{\partial}{\partial\xi}\left(\frac{h_3}{h_1} k_\xi \frac{\partial T}{\partial\xi}\right) - \frac{1}{x_b}\frac{\partial}{\partial\xi}\left(\frac{h_3}{h_1} k_\xi \frac{\eta A}{\delta}\frac{\partial T}{\partial\eta}\right) - \frac{\eta A k_\xi}{\delta x_b}\frac{a}{\partial\eta}\left(\frac{h_3}{h_1}\frac{\partial T}{\partial\xi}\right)\right.$$

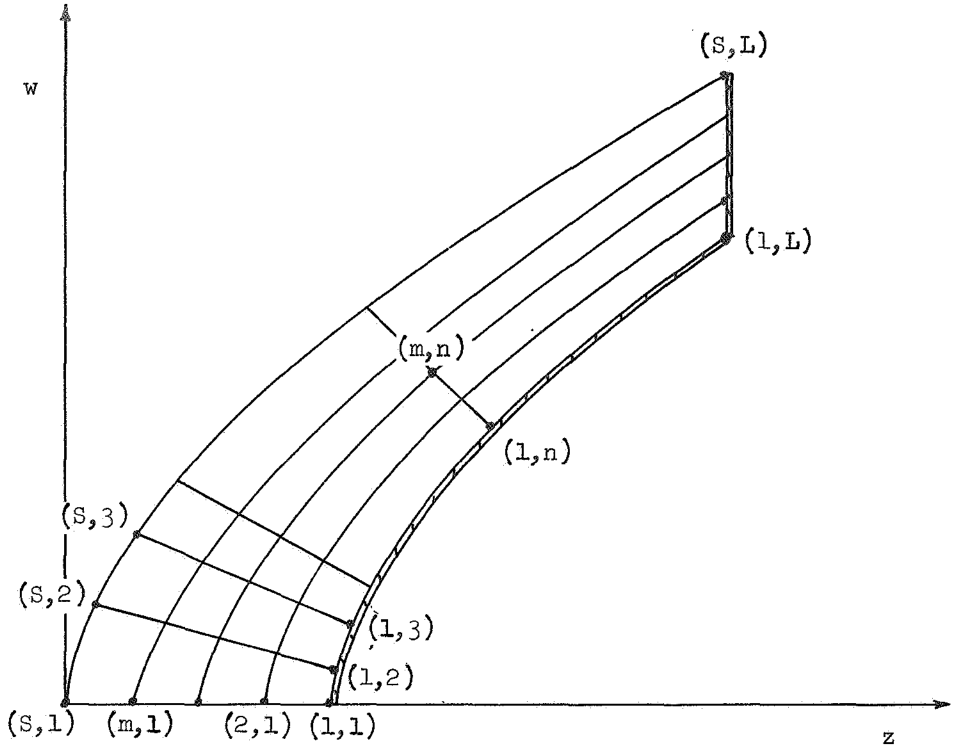
$$\left. + \frac{\eta A}{\delta^2} k_\xi \frac{\partial}{\partial\eta}\left(\frac{h_3}{h_1}\eta A \frac{\partial T}{\partial\eta}\right)\right] = \rho c_p\left(\frac{\partial T}{\partial\tau} + \frac{\dot{m}\eta}{\rho\delta}\frac{\partial T}{\partial\eta}\right) \tag{4}$$

where

$$A = \frac{1}{x_b}\frac{\partial\delta}{\partial\xi} \tag{5}$$

The unknown temperature field defined by the solution to equation (4) and its boundary condition was obtained by first approximating these equations by finite-difference equations with the use of the node pattern shown in sketch 3. Then the solution to these finite-difference equations is obtained with the method used in reference 2.

This method is classed as an alternating-direction implicit method which has the advantages of being implicit, stable, and amenable to rapid solution. This method involves the alternate use of two finite-difference analogs to equation (1). In the first finite-difference equation at time $\tau$ the analog to one of the second derivatives $\frac{\partial^2 T}{\partial x^2}$, for example, is written at the new time $\tau + \Delta\tau$, and the analog to the other derivative $\frac{\partial^2 T}{\partial y^2}$ is written at the old time $\tau$. Therefore, this equation is implicit in the x-direction (row) and explicit in the y-direction (column).




Sketch 3

In the second finite-difference equation, at time $\tau + 2\Delta\tau$, the analog $\frac{\partial^2 T}{\partial y^2}$ is written at the new time $\tau + 2\Delta\tau$ and the analog to $\frac{\partial^2 T}{\partial x^2}$ is written at the old time $\tau + \Delta\tau$. The second equation is implicit in the y-direction (column) and explicit in the x-direction (row). Using the two equations alternately results in a stable solution for any ratio of time increment to space increment as long as the same time increment is used for the successive application of the two equations. The time increment may be changed after the successive application of the equations.

Equation (4) and the boundary conditions, when approximated by finite differences, lead to L sets of S simultaneous equations for a column solution and S sets of L simultaneous equations for a row solution. These equations take the form

$$\left.\begin{aligned} B_1T_1 + C_1T_2 &= D_1 \\ A_jT_{j-1} + B_jT_j + C_jT_{j+1} &= D_j \qquad (2 \leqq j \leqq (N-1)) \\ A_{N-1}T_{N-1} + B_NT_N &= D_N \end{aligned}\right\} \qquad (6)$$

where N is equal to S or L depending upon which finite-difference analog is applied.

Since the coefficients of equations (6) form a tridiagonal matrix, this set of simultaneous equations can be quickly solved for temperatures. The method of solution based on the Gauss elimination method is discussed in reference 3.

The coefficients of equations (6) are temperature dependent. Therefore, an iteration on these coefficients is made to obtain a temperature solution.

## OPERATION OF PROGRAM

The physical problem to be modeled with the analysis is described by the FORTRAN input variables listed in a subsequent section. For example, the external body geometry is described in the w,z coordinates (sketch 2) which correspond to the input variables RS and ZS; material density corresponds to the input variable RO; and the stagnation cold-wall heating rate corresponds to the input variable QCTAB, which is a time-dependent array. Other input variables are required which control the solution, specify boundary conditions, and determine output from the program. These variables are listed in a subsequent section.

This section describes the various boundary conditions that are available and a plotting routine that may be used with the output. The computation of the computing interval is also discussed.

### Boundary Conditions Along Front Surface

An energy balance at the surface is

$$q_C\left(1 - \frac{H_w}{H_e}\right)\left\{1 - (1 - \beta)\left[0.6\,\frac{H_e}{q_C}\left(\alpha_c\dot{m}_c + \alpha_s\dot{m}_s\right) - 0.084\left(\frac{H_e}{q_C}\right)^2\left(\alpha_c\dot{m}_c + \alpha_s\dot{m}_s\right)^2\right]\right.$$
$$\left. - \beta\left(\alpha_c\dot{m}_c + \alpha_s\dot{m}_s\right)\left(\frac{H_e}{q_C}\right)\right\} + \alpha q_r + \dot{m}_c\Delta H_c = k_y\,\frac{\partial T}{\partial y} + \dot{m}_s\Delta H_s + \sigma\epsilon T_w^4 \qquad (7)$$

where the terms on the left of the equality sign represent energy input to the surface and the terms on the right represent energy dissipation at the surface. The energy input may be any combination of convective heating, radiant heating, and the heat resulting from combustion.

This energy input is accommodated by the heat conducted away from the surface and any combination of the heat radiated from the surface and the heat absorbed by sublimation. The quantity of energy involved in each process is specified by the values assigned to the FORTRAN variables associated with that process. For example, the

FORTRAN variables associated with the radiant heating rate $q_r$ are QRTAB, ALPHAT, and QRRAT, all of which define the radiant heating to the body with time.

The pressure and the convective and radiant heating rates are functions of the body shape and also vary over the body surface. The changes in $q_C$ and $q_r$ at the stagnation point and the changes in pressure, $q_C$, and $q_r$ around the body are computed within the program by setting IADJUST to a value greater than zero and specifying values for the variables defining the flow field and the body geometry. If IADJUST equals zero, then the variation of $q_C$, $q_r$, and the pressure over the body are tabulated as QRAT, QRRAT, and PRAT, respectively.

Equation (7) shows that the mass loss due to combustion $\dot{m}_c$ and mass loss due to sublimation $\dot{m}_s$ affect the energy balance. This effect can be specified by either transpiration theory ($\beta = 0$) or linear ablation theory ($\beta = 1$).

The rates of mass loss by both oxidation and sublimation are computed at each time step. However, only the larger of the two is used.

The rate of mass loss by combustion may be specified by a half-order or a first-order oxidation equation. The input XORDER specifies which equation is used. The equation for a half-order oxidation reaction is (eq. (15) in ref. 1)

$$\dot{m}_c = \frac{1}{2}\left\{-\frac{M_w(H_e - H_w)K^2 p_w}{M_{O_2} q_{C,net}\lambda} + \sqrt{\left[\frac{M_w(H_e - H_w)K^2 p_w}{M_{O_2} q_{C,net}\lambda}\right]^2 + 4K^2 C_e \frac{M_w}{M_{O_2}} p_w}\right\} \tag{8}$$

where

$$q_{C,net} = q_C\left(1 - \frac{H_w}{H_e}\right)\left\{1 - (1 - \beta)\left[0.6\frac{H_e}{q_C}(\alpha_c\dot{m}_c + \alpha_s\dot{m}_s) - 0.084\left(\frac{H_e}{q_C}\right)^2(\alpha_c\dot{m}_c + \alpha_s\dot{m}_s)^2\right] - \beta(\alpha_c\dot{m}_c + \alpha_s\dot{m}_s)\left(\frac{H_e}{q_C}\right)\right\} \tag{9}$$

and

$$K = A_c e^{-B_c/T_w} \tag{10}$$

The equation for a first-order oxidation reaction is (eq. (16) in ref. 1)

$$\dot{m}_C = \frac{Kp_w C_e}{\frac{M_{O_2}}{M_w} + \frac{Kp_w(H_e - H_w)}{q_{C,net}{}^{\lambda}}} \tag{11}$$

The rate of mass loss by sublimation is (eq. (17) in ref. 1)

$$\dot{m}_S = \frac{A_s(p_w)^p}{\left(R_{stag}\right)^r} e^{-B_s/T_w} \tag{12}$$

Boundary Conditions Along Back Surface and Edge of Body

Several boundary conditions may be specified along the surfaces at $y = 0$ and $x = x_b$. These conditions are a constant-property heat sink, radiation from these surfaces to a surface at a specified temperature, or any combination of these. A heat sink along the back of the body is specified by the inputs CPDP, RODP, TDPRIME; along the edge of the body, by CPP, ROP, and TPRIME. Radiation from these surfaces is specified by the inputs EPSONPP, EPSONEP, and TBTAB.

Output Plotting Routine

The plotting routine for this program is convenient for studying the results of calculations. This routine is activated by setting IPLOT equal to an integer greater than zero. The following plots are generated: (1) RSS versus ZS at times listed in the PLTIME table (this plot shows the body geometry), (2) MDOT versus X at each PRFREQ time, and (3) T(N) versus X at each PRFREQ time, where N is a specified row of temperatures. For example, to plot the temperatures of rows 2, 6, and 8, set the input NTP = 3, 2, 6, 8, where the 3 specifies the number of rows to be plotted. Other input quantities that must be specified are MDMAX, RSSMAX, ZSMAX, PTMAX, and PTMIN. These inputs specify maximum and minimum values which are used to get reasonable plotting scales. Sample plots are shown with example problems discussed in a subsequent section.

The plotting routines used are from the CalComp software package. Plotter output is routed to a tape during job execution and after job completion is plotted on a CalComp digital incremental plotter.

Computing Interval

Although the alternating-direction implicit method used for solution of the finite-difference equations has the advantage of being stable for any time increment, the choice

of a computing interval is important. An initial and a maximum computing interval DELTAU and DTMAX are inputs for the program. After the application of a column and a row solution, the program computes an interval for the next two successive time steps. This is done by examining the number of iterations necessary for convergence at the previous time step. If this number was (a) equal to 1, the computing interval will be doubled, but will not exceed DTMAX; (b) equal to 2, the interval will not be changed; or (c) equal to 3, the interval will be halved.

This should not be confused with the input MAXITT. If the number of iterations during a solution that is not a row solution exceeds MAXITT, the computing interval will be halved and the solution restarted.

## PROGRAM DESCRIPTION

The computer program D2430 was written in FORTRAN IV language for the Control Data 6000 series digital computer under the SCOPE 3.0 operating system. The program requires approximately 70 000 octal locations of core storage.

This section presents the program, its subroutines, and their variables. The variables are grouped in labeled COMMON blocks PICK, INPUTS, and HOLD. Input data are loaded with FORTRAN IV NAMELIST. The variables in INPUTS (except the variable DUMMY) and in HOLD are also in the NAMELIST statement which appears in another section.

### Labeled COMMON

The following list contains the FORTRAN variables appearing in labeled COMMON and the dimensions of the array for each variable. The notation is in the form A(m,n).

| COMMON label | FORTRAN variable | Description |
|---|---|---|
| PICK | | |
| | A(10,20) | Elements in coefficient matrix for the column solution |
| | AA(20) | $\frac{\partial \delta}{\partial x}$ |
| | AB(10,20) | Elements in coefficient matrix for the row solution |
| | ALPHA(20) | $\alpha$ |
| | B(20) | Major diagonal elements in coefficient matrix |

| COMMON label | FORTRAN variable | Description |
|---|---|---|
| PICK | | |
| | BS1(10,20) | Major diagonal elements in coefficient matrix for the column solution minus $\frac{\partial T}{\partial \tau}$ term |
| | BS1B(10,20) | Major diagonal elements in coefficient matrix for the row solution minus $\frac{\partial T}{\partial \tau}$ term |
| | C(10,20) | Elements in coefficient matrix for the column solution |
| | CB(10,20) | Elements in coefficient matrix for the row solution |
| | CK(10) | Temporary storage used to define the thermal conductivity at a half station |
| | CKETA(10,20) | $k_\eta$ at the station |
| | CKXI(10,20) | $k_\xi$ at the station |
| | COST(20) | $\cos\theta$ |
| | CP(10,20) | $c_p$ |
| | D(10,20) | $\frac{h_2 h_3 k_\xi}{h_1}$ |
| | DC(20) | Right-hand side of the matrix solution |
| | DELESQ | $(\Delta\eta)^2$ |
| | DELETA | $\Delta\eta$ |
| | DELTA(20) | $\delta$ |
| | DELXI | $\Delta\xi$ |
| | DELXISQ | $(\Delta\xi)^2$ |
| | E(10,20) | $\frac{h_1 h_3 k_\eta}{h_2}$ |
| | EIGHT3 | Constant, 8.0/3.0 |
| | ELAM(20) | $\lambda$ |
| | ETA(10) | $\eta$ |

| COMMON label | FORTRAN variable | Description |
|---|---|---|
| PICK | | |
| | EXPG | Computed constant used in computing new heating distribution |
| | F(10,20) | $\frac{h_2 h_3 k_\xi \eta}{h_1 \delta} \frac{\partial \delta}{\partial x}$ |
| | GG | Computed constant used in computing new heating distribution |
| | GIMACH | Computed constant used in computing new pressure distribution |
| | H1(10,20) | $h_1$ |
| | H2(10,20) | $h_2$ |
| | H3(10,20) | $h_3$ |
| | HC(20) | $\Delta H_s$ |
| | HCOMB(20) | $\Delta H_c$ |
| | HE | $H_e$ |
| | HW(20) | $H_w$ |
| | IFIRST | Internal code; 0 for first time step in calculation, 1 for any time after first time step |
| | IROCOL | Internal code; 1 for column solution, 2 for row solution |
| | ITC | Number of iterations during the column solution |
| | ITR | Number of iterations during the row solution |
| | ITT | Number of iterations during a solution |
| | ITTO | Total number of iterations from the initial time |
| | LM1 | Computed constant (L-1) |
| | LM2 | Computed constant (L-2) |
| | MCDOT(20) | $\dot{m}_c$ |
| | MDOT(22) | $\dot{m}$ |
| | MSDOT(20) | $\dot{m}_s$ |

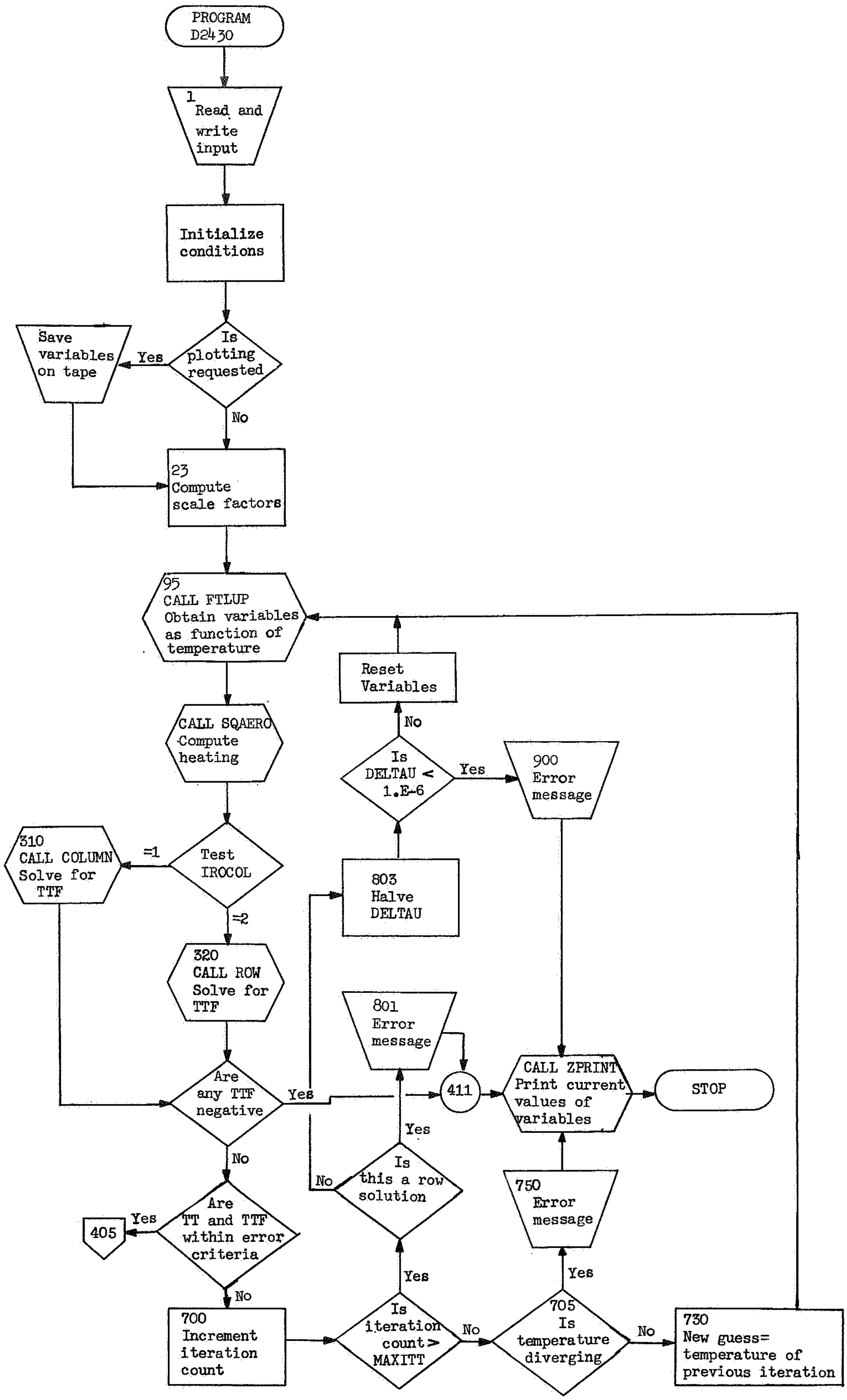
| COMMON label | FORTRAN variable | Description |
|---|---|---|
| PICK | | |
| | PID2 | Constant 1.5707963268 |
| | PRELOC(20) | Local wall pressure |
| | QC(20) | Adjusted convective heating rate |
| | QC1 | $q_C$ |
| | QCNET | $q_{C,net}$ |
| | QCOMB(20) | Heat due to combustion for oxidation |
| | QR(20) | Adjusted radiant heating rate |
| | QR1 | $q_r$ |
| | QS(20) | Net heat input |
| | RNS | Nose radius |
| | RODPC | $t''\rho''c_p''/\Delta\tau$ |
| | ROPCPP | $t'\rho'c_p'/\Delta\tau$ |
| | RSS(22) | Coordinate used to define body geometry, w |
| | RSTO2 | Computed constant, ratio of molecular weight of free stream to molecular weight of diatomic oxygen used in oxidation equation |
| | SIG | Computed constant $\sigma\epsilon$ |
| | SIGDP | Computed constant $\sigma\epsilon''$ |
| | SIGMA | $\sigma$ |
| | SIGP | Computed constant $\sigma\epsilon'$ |
| | SINT(20) | sin $\theta$ |
| | SM1 | Computed constant (S-1) |
| | SM2 | Computed constant (S-2) |
| | TAU | Time at which calculation is being made |
| | TB | Temperature to which back surfaces radiate |
| | TT(10,20) | Estimated temperatures at $\tau$ |
| | TWDELXI | Computed constant $2\ \Delta\xi$ |

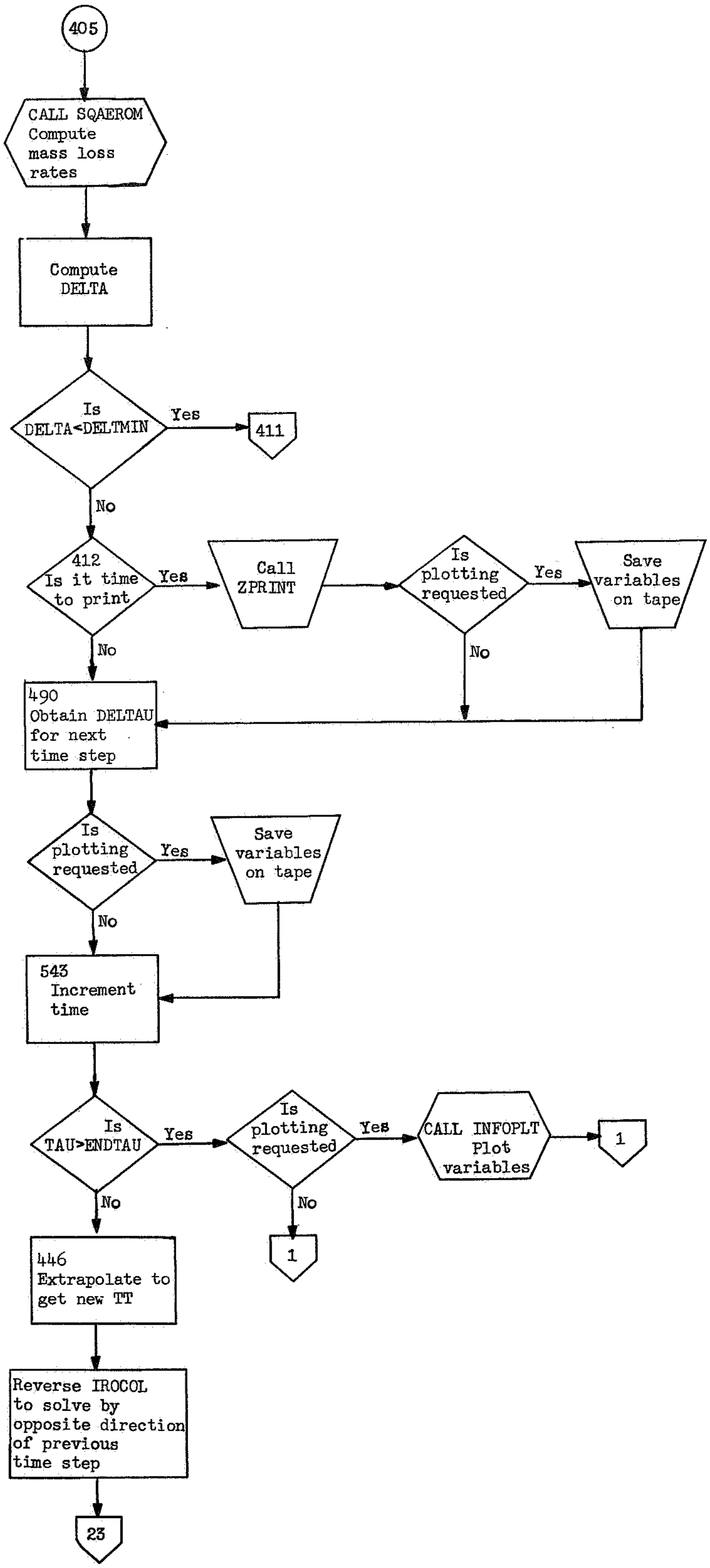
| COMMON label | FORTRAN variable | Description |
|---|---|---|
| PICK | | |
| | TWOGI | Computed constant used in computing new heating distribution |
| | V(20) | Elements in coefficient matrix for column solution |
| | VB(10) | Elements in coefficient matrix for row solution |
| | X(22) | Curvilinear coordinate |
| | XDXISQ | Computed constant $(x_b\ \Delta\xi)^2$ |
| | XODXI | Computed constant $x_b\ \Delta\xi$ |
| | Y(10,20) | Curvilinear coordinate |
| | Z(20) | Elements in coefficient matrix for the column solution |
| | ZB(10) | Elements in coefficient matrix for the row solution |
| INPUTS | DUMMY | Used in setting initial values of all inputs to zero |
| | All the variables in NAMELIST except TMIN also appear in INPUTS | |
| HOLD | TMIN | A minimum temperature value |

## Descriptions, Flow Charts, and Listings

This section identifies the main program and each subroutine in the program D2430. A brief discussion, a flow chart, and a listing for each are given. The numbers appearing in the flow charts represent a FORTRAN statement number in the program. The interpolation subroutines FTLUP and DISCOT are described in detail in appendix A.

Program D2430.- Program D2430 is the control program. It reads the inputs, calls the subroutines to solve for the temperature profile, calls subroutines for plotting, and controls the iteration scheme for the temperature solution. The flow chart for program D2430 is given on the following pages:


PROGRAM D2430
1 Read and write input
Initialize conditions
Is plotting requested
Yes
No
Save variables on tape
23 Compute scale factors
95 CALL FTLUP Obtain variables as function of temperature
Reset Variables
No
Is DELTAU < 1.E-6
Yes
900 Error message
CALL SQAERO Compute heating
Test IROCOL
=1
=2
310 CALL COLUMN Solve for TTF
320 CALL ROW Solve for TTF
803 Halve DELTAU
801 Error message
Are any TTF negative
Yes
No
411
CALL ZPRINT Print current values of variables
STOP
Is this a row solution
Yes
No
750 Error message
Are TT and TTF within error criteria
Yes
No
405
700 Increment iteration count
Is iteration count > MAXITT
No
Yes
705 Is temperature diverging
Yes
No
730 New guess= temperature of previous iteration


405
CALL SQAEROM Compute mass loss rates
Compute DELTA
Is DELTA<DELTMIN
Yes
411
No
412 Is it time to print
Yes
Call ZPRINT
Is plotting requested
Yes
Save variables on tape
No
No
490 Obtain DELTAU for next time step
Is plotting requested
Yes
Save variables on tape
No
543 Increment time
Is TAU>ENDTAU
Yes
Is plotting requested
Yes
CALL INFOPLT Plot variables
1
No
1
No
446 Extrapolate to get new TT
Reverse IROCOL to solve by opposite direction of previous time step
23

The listing for program D2430 is as follows:

```
      PROGRAM D2430 (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE7=201,
     1TAPE8=201,TAPE9=201)
C
C AXISYMMETRIC ABLATION PROGRAM
C TWO-DIMENSIONAL ABLATION ANALYSIS FOR AXIALLY SYMMETRIC BODIES OF REVOLUTION
C AT HIGH HEATING RATES, CONSIDERING SHAPE CHANGE
C
C THIS IS THE MAIN PROGRAM - IT CONTROLS THE GENERAL FLOW OF PROGRAM
C
      COMMON /PICK/ A(10,20),AA(20),AB(10,20),ALPHA(20),B(20),
     2 BS1(10,20),BS1B(10,20),C(10,20),CB(10,20),CK(10),CKETA(10,20),
     4 CKXI(10,20),COST(20),CP(10,20),D(10,20),DC(20),
     6 DELESQ,DELETA,DELTA(20),DELXI,DELXISQ,E(10,20),EIGHT3,
     8 ELAM(20),ETA(10),EXPG,F(10,20),GG,GIMACH,H1(10,20),H2(10,20),
     A H3(10,20),HC(20),HCOMB(20),HE,HW(20),IFIRST,IROCOL,ITC,ITR,ITT,
     C ITTO,LM1,LM2,MCDOT(20),MDOT(22),MSDOT(20),PID2,PRELOC(20),QC(20),
     E QC1,QCNET(20),QCCMB(20),QR(20),QR1,QS(20),RNS,RODPC,ROPCPP,
     G RSS(22),RSTO2,SIG,SIGDP,SIGMA,SIGP,SINT(20),SM1,SM2,TAU,TB,
     I TT(10,20),TTF(10,20),TWDELXI,TWOGI,V(20),VB(10),X(22),XDXISQ,
     K XCDXI,Y(10,20),Z(20),ZB(10)
      COMMON /INPUTS/ DUMMY(1),AEXP,ALCTAB(10),TTALC(10),MALPHC,NALPHC,
     2 ALPHAT(10),TALPHA(10),MALPHA,NALPHA,ALSTAB(10),TTALS(10),MALPHS,
     4 NALPHS,ASEXP,BETA,BEXP,BSEXP,CE,CKETATB(50),TTCKETA(10),
     6 ETATAB(5),NCKETA,NETA,CKXITAB(50),TTCKXI(10),XITAB(5),NCKXI,
     8 NXI,CORDSY,CPDP,CPP,CPTAB(10),TTABCP(10),MCP,NCP,DELTAO(20),
     A DELTAU,DELTMIN,DTMAX,ELAMTB(28),TTELAM(7),PELAM(4),NELAM,
     C NPELAM,ENDTAU,EPSCNE,EPSONEP,EPSCNPP,ERRORT,GAMBAR,GAMINF,
     E HCOMBTB(28),TTHCOMB(7),PHCOMB(4),NHCOMB,NPHCCMB,HCTAB(28),
     G TTABHC(7),PHC(4),NHC,NPHC,HETAB(10),TTABHE(10),MHE,NHE,HWTAB(15),
     I TTABHW(15),MHW,NHW,IADJUST,IPLOT,L,MACHNO,MAXITT,MDMAX,
     J MDCTO(20),
     K MWO2,MWSTR,NTP(7),PLTIME(15),PRAT(20),PRFREQ,PSEXP,PSTAGTB(10),
     M TTPSTAG(10),MPSTAG,NPSTAG,PTMAX,PTMIN,QCTAB(10),TTABQC(10),MQC,
     N NQC,QRAT(20),
     O QRRAT(20),QRTAB(10),TTABQR(10),MQR,NQR,R(20),RIEXP,RNSI,RO,RODP,
     Q ROP,RS(20),RSSMAX,S,STEBCL,T(10,20),TAUC,TBTAB(10),TTABTB(10),
     S MTB,NTB,TDPRIME,THETA(20),TPRIME,XO,XCRDER,ZS(20),ZSMAX
      DIMENSION DELT(10,20),ZZ(22),Y3L(2)
      REAL MDOTC,MDOT,MCDOT,MSDOT,MWSTR,MWO2,MACHNO,MDMAX
      INTEGER S,SM1,SM2
      DATA XLABEL,YLABEL,X2L,Y2L,Y3L/ 2HZB,3HRSS,1HX,4HMDOT,12HTEMPERATU
     1RES/
      NAMELIST /D2430/       AEXP,ALCTAB,TTALC,MALPHC,NALPHC,ALPHAT,
     2 TALPHA,MALPHA,NALPHA,ALSTAB,TTALS,MALPHS,NALPHS,ASEXP,
     4 BETA,BEXP,BSEXP,CE,CKETATB,ETATAB,TTCKETA,NCKETA,NETA,
     6 CKXITAB,XITAB,TTCKXI,NCKXI,NXI,CORDSY,CPDP,CPP,CPTAB,TTABCP,MCP,
     8 NCP,DELTAO,DELTAU,DELTMIN,DTMAX,ELAMTB,TTELAM,PELAM,NELAM,NPELAM,
     9 ENDTAU,EPSCNE,EPSONEP,EPSCNPP,ERRORT,GAMBAR,GAMINF,HCCMBTB,
     A TTHCOMB,PHCOMB,NHCOMB,NPHCCMB,HCTAB,TTABHC,PHC,NHC,NPHC,HETAB,
     C TTABHE,MHE,NHE,HWTAB,TTABHW,MHW,NHW,IADJUST,IPLOT,L,MACHNO,
     E MAXITT,MDMAX,MDOTO,MWO2,MWSTR,NTP,PLTIME,PRAT,PRFREQ,PSEXP,
     G PSTAGTB,TTPSTAG,MPSTAG,NPSTAG,PTMAX,PTMIN,QCTAB,TTABQC,MQC,
     I NQC,QRAT,QRRAT,QRTAB,TTABQR,MQR,NQR,R,RIEXP,RNSI,RO,RODP,ROP,RS,
     K RSSMAX,S,STEBCL,T,TAUO,TBTAB,TTABTB,MTB,NTB,TDPRIME,THETA,
     M TMIN,TPRIME,XC,XORDER,ZS,ZSMAX
      COMMON /HOLD/ TMIN
      TMIN =0.
      DO 10 I=1,934
   10 DUMMY(I)=0.0
      DTMAX=2.
    1 READ (5,100)
  100 FORMAT (80H
     1                                                   )
      IF (EOF,5) 2,3
    2 STOP
    3 READ (5,D2430)
      WRITE (6,D2430)
      WRITE (6,100)
C
C SET INITIAL VALUES
```

```
C
      NNTP= NTP(1)
      PID2 = 1.5707963268
      TWO GI  = 2.0 /((GAMINF - 1.0) * MACHNO **2)
      EXPG =(GAMBAR - 1.0)/ GAMBAR
      GIMACH=    1./(GAMINF * MACHNO **2)
      GG= SQRT( EXPG * (1.0 + TWOGI) * (1.0- GIMACH))
      GG= SQRT (GG) * 2.0
      INCP=0
      IROW=0
      IDT=1
      DTAU0=1.0
      DTAU1=DELTAU
      IROCOL =1
C WILL PRINT ONLY AFTER A COL. AND ROW COMPUTATION HAS BEEN MADE
      TAU00= TAU0+ PRFREQ
      ITT0=0
      DO 11 M=1,S
      DO 11 N=1,L
      DELT(M,N)=1000.
   11 TT(M,N)= T(M,N)
      DELTAU=DELTAU/2.
      TAU=TAU0+DELTAU
      IFIRST=0
      ITT=1
      LM1 = L- 1
      ALM1 = L M1
      LM2 = L- 2
      SM1 = S- 1
      SM2 = S- 2
      DELXI =1./ALM1
      DELX =X0/ALM1
      RST02 = MWSTR/MW02
      X(1) =0.
      DO 12 N=2,L
   12 X(N) = X(N-1) + DELX
      DELETA = 1./SM1
      DELXISQ = DELXI **2
      DELESQ= DELETA **2
      TWDELXI = 2.0* DELXI
      EIGHT3=8.0/3.0
      DO 18 M=1,S
      AM=M-1
   18 ETA(M)=DELETA*AM
      SIGMA=STEBOL
      SIG = SIGMA* EPSONE
      SIGP = SIGMA * EPSONEP
      SIGDP= SIGMA * EPSONPP
      X0DXI = X0 * DELXI
      RODPC  = TDPRIME*RODP * CPDP / DELTAU
      ROPCPP = TPRIME * ROP * CPP/ DELTAU
      RODT= R0/DELTAU
      XDXISQ = X0**2 * DELXISQ
      DO 22 N=1,L
      MDOT(N)=MDOT0(N)
      MCDOT(N)=MDCTO(N)
      MSDOT(N)=MDOT0(N)
   20 DELTA(N)= DELTA0(N)
      THETA(N)=.0174532925*THETA(N)
      SINT(N) = SIN(THETA(N))
      ZZ(N)= ZS (N)+DELTA0(N)*SINT(N)
   22 COST(N)= COS(THETA(N))
      IF (IPLOT.EQ.0) GO TO 23
C PLOT BASE CURVE IF PLOTTING IS CALLED FOR
      REWIND 7
      REWIND 8
      REWIND 9
```

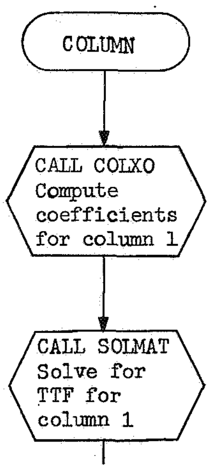
```
      CALL CALCCMP
      IPLT=1
      IPLTK=0
      IF (CORDSY.NE.0) GO TO 2250
      WRITE   (7) (ZZ(N),RS(N),N=1,L)
      GO TO 23
 2250 WRITE (7) (ZS(N),DELTA(N),N=1,L)
C
C COMPUTE  H-S
C
   23 DO 25 M=1,S
      DO 25 N=1,L
      Y(M,N)=ETA(M)*DELTA(N)
      H1(M,N) = 1.0 + ETA(M)* DELTA(N)/R(N)
      H2(M,N)=1.
   25 H3(M,N)= RS(N) + Y(M,N) *COST(N)
   95 DO 101 M=1,S
      DO 101  N=1,L
      CALL FTLUP (TT(M,N),CP(M,N),MCP,NCP,TTABCP,CPTAB)
      CALL DISCCT (TT(M,N),X (N),TTCKXI   ,CKXITAB,XITAB,11,NCKXI,NXI,
     1CKXI(M,N))
  101 CALL DISCCT(TT(M,N),Y(M,N),TTCKETA ,CKETATB,ETATAB,11,NCKETA,NETA,
     2CKETA(M,N))
      AA(1)=0.0
      DO 109 N=2,LM1
  109 AA(N)= (DELTA(N+1)-DELTA(N-1))/(TWDELXI*XO)
      AA(L)=(3.0*DELTA(L)-4.0*DELTA(LM1)+DELTA(LM2))/(TWDELXI*XO)
      DO 110 N=1,L
      DO 110 M=1,S
      D(M,N)=    H2(M,N)*H3(M,N)* CKXI(M,N)/H1(M,N)
      E(M,N)=    H1(M,N)* H3(M,N) * CKETA(M,N) / H2(M,N)
  110 F(M,N)=D(M,N)*ETA(M)*AA(N)/DELTA(N)
      CALL SQAERC
      GO TO (310,320), IROCOL
  310 CALL COLUMN
      ITC=ITT
      IFIRST=1
      GO TO 350
  320 CALL ROW
      ITR=ITT
      IF (IROW.EQ.0) IROW=2
  350 CONTINUE
C IF ANY TEMPERATURES ARE NEGATIVE  STOP  CALCULATIONS
      DO 360 N=1,L
      DO 360 M=1,S
      IF (TTF(M,N).LE.0) GO TO 411
  360 CONTINUE
C TEST TO SEE IF TEMPERATURES HAVE CONVERGED
C
      ITTO=ITTO+1
      DO 400 N=1,L
      DO 400 M=1,S
      ABSTT=ABS(TT(M,N))
      ABSTTF=ABS(TTF(M,N))
      TEST=ABS(ABSTTF-ABSTT)/ABSTT
      IF (TEST - ERRCRT) 400,400,700
  400 CONTINUE
C
C COMPUTE MDOT
C
      CALL SQAEROM
C COMPUTE DELTA
      DO 410 N=1,L
      DELTA(N)=DELTAO(N)-(MDOTO(N)+MDOT(N))*DELTAU/(2.0*RO)
C RESET DELTAO AND MDOTO
  410 MDOTO(N)=MDOT(N)
C IF DELTA BECOMES LESS THAN DELTMIN (SOME MINIMUM DELTA INPUT) STOP
```

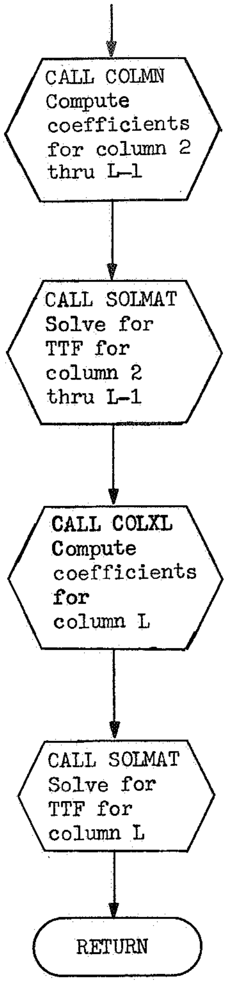
```
C THE CALCULATIONS
      DO 412 N=1,L
      IF (DELTA(N).GT. DELTMIN) GO TO 412
  411 CALL ZPRINT
      STOP
  412 CONTINUE
      IF (INOP.EQ.1) GO T O 418
      IF (TAU.LT.TAUOO) GO TO 420
      IF (IROCOL.EQ.1) GO TO 418
      INOP=1
      GO TO 420
  418 INOP =0
      TAUOO=TAUOO+ PRFREQ
C
C
      CALL ZPRINT
C
      IF (IPLOT.EQ.0) GO TO 420
      IPLTK= IPLTK + 1
      WRITE(8) (      MDOT(N), N=1,L)
      IF (NNTP.EQ.0) GO TO 420
      DO 419 M=1,NNTP
      I= NTP(M+1)
  419 WRITE (9)  (TTF(I,N),N=1,L)
  420 IF (IROW-1) 540,490,484
  484 DELTAU=DELTAU*2.0
      IROW=1
      KFRE=KFRE+1
C
C OBTAIN DELTAU AS A FUNCTION OF ITERATION OF PREVIOUS TIME STEP
  490 DTAU1 = DELTAU
      IF (IROCOL.EQ.1) GO TO 540
      IF (ITT-2) 495,540,530
  495 DELTAU=2.0*DTAU1
      IF (DELTAU.GT.DTMAX) DELTAU=DTMAX
      GO TO 540
  530 DELTAU=DTAU1/2.
      IF (DELTAU.LT.1.E-6) GO TO 900
  540 TAUO = TAU
C CHECK TO SEE IF IT IS TIME TO PLOT
      IF (IPLOT.EQ.0) GO TO 543
      IF (TAU.LT.PLTIME(IPLT)) GO TO 543
      IPLT=IPLT+1
      IF (CORDSY.NE.0) GO TO 542
      WRITE (7) (ZS(N),RSS(N),N=1,L)
      GO TO 543
  542 WRITE (7) (ZS(N),DELTA(N),N=1,L)
C
C INCREMENT TIME AND REPEAT CYCLE ALTERNATING ROW AND COLUMN SOLUTION
  543 TAU=TAU+DELTAU
      RODPC  = TDPRIME*RODP * CPDP / DELTAU
      ROPCPP = TPRIME * ROP * CPP/ DELTAU
      RODT= RO/DELTAU
      IF (TAU.GT.ENDTAU)  GO TO 950
C
C EXTRAPOLATE TO GET NEW GUESS TEMP(TT)
C
      DO 446 M=1,S
      DO 446 N=1,L
      DELT(M,N)=1000.
      DELTN=TTF(M,N)-T(M,N)
      T(M,N)=TTF(M,N)
  446 TT(M,N)=TTF(M,N)+(DELTAU/DTAU1)*DELTN
      GO TO (550,650),IROCOL
  550 IROCOL = 2
      ITT=1
      GO TO 23
```

```
  650 IROCOL = 1
      ITT=1
      GO TO 23
C
C TEMP. DOES NOT MEET ERROR CRITERIA,  MUST ITERATE AGAIN
C NEW GUESS IS TEMP. OF  PREVIOUS ITERATION   TT =TTF
C
  700 ITT =ITT +1
      IF (ITT - MAXITT) 705,705,800
  705 DO 720 N=1,L
      DO 720 M=1,S
      DELT1 = ABS(TTF(M,N)- TT(M,N))
      IF (DELT1.LT.10.) GO TO 718
      IF (DELT1 -DELT(M,N)) 718,750,750
  718 DELT(M,N)=DELT1
  720 CONTINUE
      DO 730 M=1,S
      DO 730 N=1,L
  730 TT(M,N)= TTF(M,N)
      GO TO 95
  750 IF (ITT.LT.3) GO TO 718
C
C PROGRAMED STOPS
C
      WRITE (6,752)
  752 FORMAT (*OTEMPERATURE IS DIVERGING ----- WHY*)
  758 WRITE (6,759)
  759 FORMAT (*OTT(M,N)*)
      DO 765 M=1,S
      MM=S-(M-1)
  765 WRITE (6,766) ETA(MM),(TT(MM,N),N=1,L)
  766 FORMAT (F6.3,6X15F8.1/(12X,15F8.1))
      WRITE (6,767) IROCOL
  767 FORMAT (*OIROCOL=*I3)
      CALL ZPRINT
      STOP
  800 IF (IROCOL.EQ.1) GO TO 803
      WRITE (6,801)
  801 FORMAT (*OTHIS IS A ROW SOLUTION, DELTAU CANNOT CHANGE)
      GO TO 758
C
  803 DTAU1= DELTAU
      DELTAU = DELTAU/2.0
      WRITE (6,805) DELTAU ,TAU
  805 FORMAT (*OI DID IT--       DELTAU=*E14.5,*TAU=*E14.5)
      IF (DELTAU. LT. 1.E-6)  GO TO 900
      TAU = TAU - DELTAU
      DO 810 M=1,S
      DO 810 N=1,L
      DELT(M,N)=1000.
  810 TT(M,N) = T(M,N)
      ITT  = 1
      GO TO 95
  900 WRITE (6,901)
  901 FORMAT  (*OTEMPERATURE ITERATION DOES NOT CONVERGE*)
      GO TO 758
C
C PLOT  ZS  VS. RSS ,   X  VS MDOT ,  X  VS BACK SURFACE TEMPERATURE
C
```

```
950 CALL ZPRINT
    IF (IPLOT.EQ.0) GO TO 1
    END FILE 7
    END FILE 8
    END FILE 9
    REWIND  7
    REWIND 8
    REWIND 9
    IEC = 0
    DO 960 M=1,IPLT
    READ (7)  (ZZ(N), RSS(N),N=1,L)
    IF (M.EQ.IPLT )  IEC =1
960 CALL INFOPLT (IEC,L,ZZ,1,RSS,1,0.,ZSMAX,0.,RSSMAX,1.,10,XLABEL,10,
   1 YLABEL,0)
    IEC =0
    DO 970 M=1,IPLTK
    READ(8) (      MDOT(N),N=1,L)
    IF (M.EQ.IPLTK)  IEC= 1
970 CALL INFOPLT (IEC,L,X,1,MDOT,1,0.,0.,0.,MDMAX,1.,10,X2L,10,Y2L,0)
    IEC =0
    IF (NNTP.EQ.0) GO TO 1
    DO 990 M=1,IPLTK
    ISYM=10
    DO 980 I=1,NNTP
    READ (9)    (ZZ(N),N=1,L)
    IF (M.EQ.IPLTK .AND. I.EQ.NNTP)  IEC =1
    ISYM= ISYM + 1
980 CALL INFOPLT (IEC,L,X,1,ZZ,1,0.,0.,PTMIN,PTMAX,1.,10,X2L,20,Y3L,
   1 ISYM)
990 CONTINUE
    GO TO 1
    END
```

Subroutine COLUMN.- Subroutine COLUMN calls the appropriate routines to compute the coefficient for the matrix solution and to solve the tridiagonal matrix for each column of temperatures. The flow chart for subroutine COLUMN is as follows:



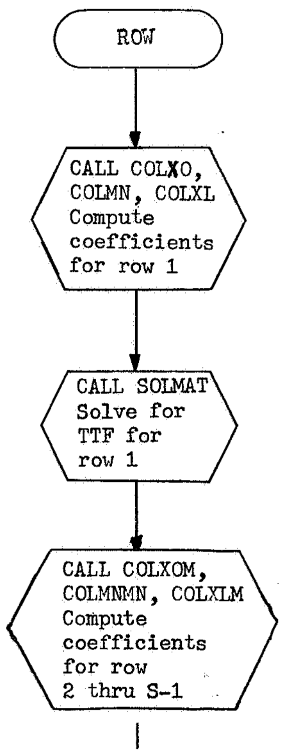


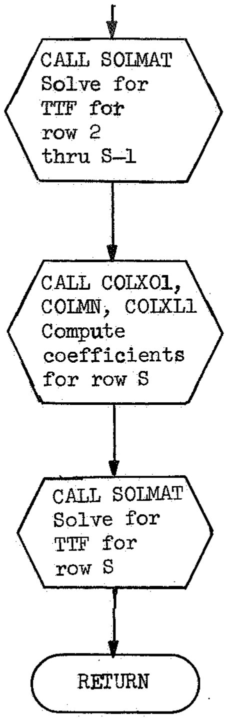

The program listing for subroutine COLUMN is as follows:

```
      SUBROUTINE COLUMN
C
C SOLVES THE MATRIX COLUMN BY COLUMN FOR ONE ITERATION
C SOLVES  M (NO. OF ROWS)  SETS OF SIMULTANEOUS EQUATICNS  N (NO. OF COLUMNS)
C TIMES            THEN  RETURNS TO MAIN PROGRAM TO TEST FOR CONVERGENCE
C
      COMMON /PICK/ A(10,20),AA(20),AB(10,20),ALPHA(20),B(20),
     2 BS1(10,20),BS1B(10,20),C(10,20),CB(10,20),CK(10),CKETA(10,20),
     4 CKXI(10,20),COST(20),CP(10,20),D(10,20),DC(20),
     6 DELESQ,DELETA,DELTA(20),DELXI,DELXISQ,E(10,20),EIGHT3,
     8 ELAM(20),ETA(10),EXPG,F(10,20),GG,GIMACH,H1(10,20),H2(10,20),
     A H3(10,20),HC(20),HCOMB(20),HE,HW(20),IFIRST,IROCOL,ITC,ITR,ITT,
     C ITTO,LM1,LM2,MCDOT(20),MDOT(22),MSDOT(20),PID2,PRELOC(20),QC(20),
     E QC1,QCNET(20),QCOMB(20),QR(20),QR1,QS(20),RNS,RODPC,ROPCPP,
     G RSS(22),RST02,SIG,SIGDP,SIGMA,SIGP,SINT(20),SM1,SM2,TAU,TB,
     I TT(10,20),TTF(10,20),TWDELXI,TWOGI,V(20),VB(10),X(22),XDXISQ,
     K XCDXI,Y(10,20),Z(20),ZB(10)
      COMMON /INPUTS/ DUMMY(1),AEXP,ALCTAB(10),TTALC(10),MALPHC,NALPHC,
     2 ALPHAT(10),TALPHA(10),MALPHA,NALPHA,ALSTAB(10),TTALS(10),MALPHS,
     4 NALPHS,ASEXP,BETA,BEXP,BSEXP,CE,CKETATB(50),TTCKETA(10),
     6 ETATAB(5),NCKETA,NETA,CKXITAB(50),TTCKXI(10),XITAB(5),NCKXI,
     8 NXI,CORDSY,CPDP,CPP,CPTAB(10),TTABCP(10),MCP,NCP,DELTAO(20),
     A DELTAU,DELTMIN,DTMAX,ELAMTB(28),TTELAM(7),PELAM(4),NELAM,
     C NPELAM,ENDTAU,EPSONE,EPSONEP,EPSONPP,ERRORT,GAMBAR,GAMINF,
```

```
     E HCOMBTB(28),TTHCCMB(7),PHCOMB(4),NHCOMB,NPHCOMB,HCTAB(28),
     G TTABHC(7),PHC(4),NHC,NPHC,HETAB(10),TTABHE(10),MHE,NHE,HWTAB(15),
     I TTABHW(15),MHW,NHW,IADJUST,IPLOT,L,MACHNO,MAXITT,MDMAX,
     J MDOTO(20),
     K MWO2,MWSTR,NTP(7),PLTIME(15),PRAT(20),PRFREQ,PSEXP,PSTAGTB(10),
     M TTPSTAG(10),MPSTAG,NPSTAG,PTMAX,PTMIN,QCTAB(10),TTABQC(10),MQC,
     N NQC,QRAT(20),
     O QRRAT(20),QRTAB(10),TTABQR(10),MQR,NQR,R(20),RIEXP,RNSI,RO,RODP,
     Q ROP,RS(20),RSSMAX,S,STEBOL,T(10,20),TAUO,TBTAB(10),TTABTB(10),
     S MTB,NTB,TDPRIME,THETA(20),TPRIME,XO,XORDER,ZS(20),ZSMAX
      REAL MDOTC,MDOT,MCDOT,MSDOT,MWSTR,MWO2,MACHNO,MDMAX
      INTEGER S,SM1,SM2
C COMPUTE  COLUMN 1
      N1 =2
      N2 =SM1
      CALL COLXO (N1,N2)
      CALL SOLMAT (A(1,1),B,C(1,1),Z(1),V(1),DC,TTF(1,1),S)
C COMPUTE  COLUMN 2 THRU LM1
      DO 300 N=2,LM1
      CALL COLMN (N1,N2,N)
      CALL SOLMAT (A(1,N),B,C(1,N),Z(N),V(N),DC,TTF(1,N),S)
  300 CONTINUE
C COMPUTE  COLUMN  L
      CALL COLXL(N1,N2)
      CALL SOLMAT (A(1,L),B,C(1,L),Z(L),V(L),DC,TTF(1,L),S)
  6)0 RETURN
      END
```

Subroutine ROW.- Subroutine ROW calls the appropriate routines to compute the coefficients for the matrix solution and to solve the tridiagonal matrix for each row of temperatures. The flow chart for subroutine ROW is as follows:



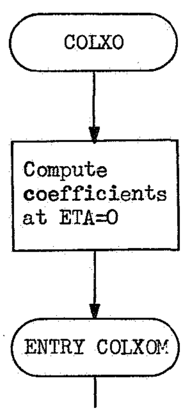


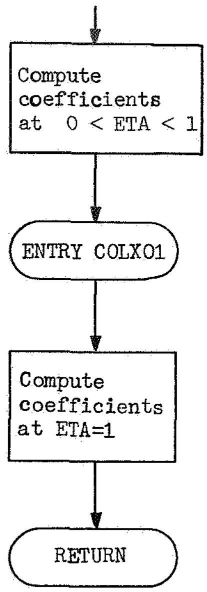

The program listing for subroutine ROW is as follows:

```
      SUBROUTINE ROW
C
C SOLVES THE MATRIX ROW BY ROW FOR ONE ITERATICN
C SOLVES  N (NO. CF CCLUMNS) SETS OF SIMULTANEOUS EQS. M(NC.CF ROWS) TIMES
C     THEN RETURNS TO MAIN PROGRAM TO CHECK FOR CONVERGENCE
C
      COMMON /PICK/ A(10,20),AA(20),AB(10,20),ALPHA(20),B(20),
     2 BS1(10,20),BS1B(10,20),C(10,20),CB(10,20),CK(10),CKETA(10,20),
     4 CKXI(10,20),COST(20),CP(10,20),D(10,20),DC(20),
     6 DELESQ,DELETA,DELTA(20),DELXI,DELXISQ,E(10,20),EIGHT3,
     8 ELAM(20),ETA(10),EXPG,F(10,20),GG,GIMACH,H1(10,20),H2(10,20),
     A H3(10,20),HC(20),HCOMB(20),HE,HW(20),IFIRST,IROCCL,ITC,ITR,ITT,
     C ITTC,LM1,LM2,MCDOT(20),MDOT(22),MSDOT(20),PID2,PRELOC(20),QC(20),
     E QC1,QCNET(20),QCOMB(20),QR(20),QR1,QS(20),RNS,RODPC,RODPCP,
     G RSS(22),RSTO2,SIG,SIGDP,SIGMA,SIGP,SINT(20),SM1,SM2,TAU,TB,
     I TT(10,20),TTF(10,20),TWDELXI,TWOGI,V(20),VB(10),X(22),XDXISQ,
     K XODXI,Y(10,20),Z(20),ZB(10)
      COMMON /INPUTS/ DUMMY(1),AEXP,ALCTAB(10),TTALC(10),MALPHC,NALPHC,
     2 ALPHAT(10),TALPHA(10),MALPHA,NALPHA,ALSTAB(10),TTALS(10),MALPHS,
     4 NALPHS,ASEXP,BETA,BEXP,BSEXP,CE,CKETATB(50),TTCKETA(10),
     6 ETATAB(5),NCKETA,NETA,CKXITAB(50),TTCKXI(10),XITAB(5),NCKXI,
     8 NXI,CORDSY,CPDP,CPP,CPTAB(10),TTABCP(10),MCP,NCP,DELTAO(20),
     A DELTAU,DELTMIN,DTMAX,ELAMTB(28),TTELAM(7),PELAM(4),NELAM,
     C NPELAM,ENDTAU,EPSONE,EPSONEP,EPSONPP,ERRORT,GAMBAR,GAMINF,
     E HCOMBTB(28),TTHCOMB(7),PHCOMB(4),NHCCMB,NPHCOMB,HCTAB(28),
     G TTABHC(7),PHC(4),NHC,NPHC,HETAB(10),TTABHE(10),MHE,NHE,HWTAB(15),
     I TTABHW(15),MHW,NHW,IADJUST,IPLOT,L,MACHNO,MAXITT,MDMAX,
     J MDOTO(20),
     K MWO2,MWSTR,NTP(7),PLTIME(15),PRAT(20),PRFREQ,PSEXP,PSTAGTB(10),
     M TTPSTAG(10),MPSTAG,NPSTAG,PTMAX,PTMIN,QCTAB(10),TTABQC(10),MQC,
     N NQC,QRAT(20),
     O QRRAT(20),QRTAB(10),TTABQR(10),MQR,NQR,R(20),RIEXP,RNSI,RO,RODP,
     Q ROP,RS(20),RSSMAX,S,STEBOL,T(10,20),TAUO,TBTAB(10),TTABTB(10),
     S MTB,NTB,TDPRIME,THETA(20),TPRIME,XO,XORDER,ZS(20),ZSMAX
      REAL MDOTC,MDOT,MCDOT,MSDOT,MWSTR,MWO2,MACHNO,MDMAX
```

```
C COMPUTE  ROW 1
      DIMENSION ANS(20), ATEMP(20), CTEMP(20)
      INTEGER SM1 ,S
      N1 =2
      N2 =LM1
      CALL COLXO (N1,N2)
      DO  300 N=2,LM1
      CALL COLMN (N1,N2,N)
  300 CONTINUE
      CALL COLXL(N1,N2)
      DO 320 N =1,L
      ATEMP(N) = AB(1,N)
 320  CTEMP(N) = CB(1,N)
      CALL SOLMAT (ATEMP,B,CTEMP,ZB(1),VB(1),DC,ANS(1),L)
      DO 400 N=1,L
  400 TTF(1,N)=ANS(N)
C COMPUTE  ROW  2  THRU SM1
      DO  600  M=2,SM1
      N1 =M
      N2 =M
      CALL COLXOM (N1,N2)
      DO 500 N=2,LM1
      CALL COLMNMN(N1,N2,N)
  500 CONTINUE
      CALL COLXLM (N1,N2)
      DO 510 N=1,L
      ATEMP(N) = AB(M,N)
  510 CTEMP(N) = CB(M,N)
      CALL SOLMAT (ATEMP,B,CTEMP,ZB(M),VB(M),DC,ANS(1),L)
      DO 590 N=1,L
  590 TTF(M,N)=ANS(N)
  600 CONTINUE
C COMPUTE ROW S
      CALL  COLXO1(N1,N2)
      DO  800 N=2,LM1
      CALL COLMN1 (N1,N2,N)
  800 CONTINUE
      CALL COLXL1(N1,N2)
      DO 810 N=1,L
      ATEMP(N) = AB(S,N)
  810 CTEMP(N) = CB(S,N)
      CALL SOLMAT  (ATEMP,B,CTEMP,ZB(S),VB(S),DC,ANS(1),L)
      DO 890 N=1,L
  890 TTF(S,N)=ANS(N)
  900 RETURN
      END
```

Subroutine COLXO.- Subroutine COLXO computes the coefficients of the tridiagonal matrix where $\xi = 0$ and $0 \leqq \eta \leqq 1$. The flow chart for subroutine COLXO is as follows:





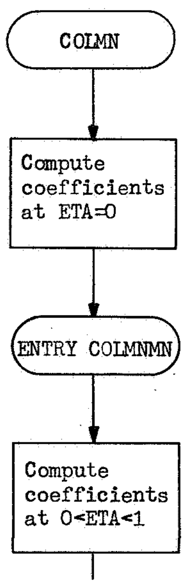

The program listing for subroutine COLXO is as follows:

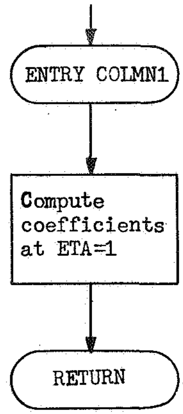
```
      SUBROUTINE COLXO(N1,N2)
C
C COMPUTE COEF. FOR XI=0,    COLUMN IMPLICIT
C IROCOL = 1        COLUMN IMPLICIT
C IROCOL = 2        ROW  IMPLICIT
C
      COMMON /PICK/ A(10,20),AA(20),AB(10,20),ALPHA(20),B(20),
     2 BS1(10,20),BS1B(10,20),C(10,20),CB(10,20),CK(10),CKETA(10,20),
     4 CKXI(10,20),COST(20),CP(10,20),D(10,20),DC(20),
     6 DELESQ,DELETA,DELTA(20),DELXI,DELXISQ,E(10,20),EIGHT3,
     8 ELAM(20),ETA(10),EXPG,F(10,20),GG,GIMACH,H1(10,20),H2(10,20),
     A H3(10,20),HC(20),HCOMB(20),HE,HW(20),IFIRST,IROCOL,ITC,ITR,ITT,
     C ITTC,LM1,LM2,MCDOT(20),MDOT(22),MSDOT(20),PID2,PRELOC(20),QC(20),
     E QC1,QCNET(20),QCOMB(20),QR(20),QR1,QS(20),RNS,RODPC,ROPCPP,
     G RSS(22),RSTO2,SIG,SIGDP,SIGMA,SIGP,SINT(20),SM1,SM2,TAU,TB,
     I TT(10,20),TTF(10,20),TWDELXI,TWOGI,V(20),VB(10),X(22),XDXISQ,
     K XODXI,Y(10,20),Z(20),ZB(10)
      COMMON /INPUTS/ DUMMY(1),AEXP,ALCTAB(10),TTALC(10),MALPHC,NALPHC,
     2 ALPHAT(10),TALPHA(10),MALPHA,NALPHA,ALSTAB(10),TTALS(10),MALPHS,
     4 NALPHS,ASEXP,BETA,BEXP,BSEXP,CE,CKETATB(50),TTCKETA(10),
     6 ETATAB(5),NCKETA,NETA,CKXITAB(50),TTCKXI(10),XITAB(5),NCKXI,
     8 NXI,CORDSY,CPCP,CPP,CPTAB(10),TTABCP(10),MCP,NCP,DELTAO(20),
     A DELTAU,DELTMIN,DTMAX,ELAMTB(28),TTELAM(7),PELAM(4),NELAM,
     C NPELAM,ENDTAU,EPSONE,EPSONEP,EPSONPP,ERRORT,GAMBAR,GAMINF,
     E HCOMBTB(28),TTHCOMB(7),PHCOMB(4),NHCOMB,NPHCOMB,HCTAB(28),
     G TTABHC(7),PHC(4),NHC,NPHC,HETAB(10),TTABHE(10),MHE,NHE,HWTAB(15),
     I TTABHW(15),MHW,NHW,IADJUST,IPLOT,L,MACHNO,MAXITT,MDMAX,
     J MDOTO(20),
     K MWO2,MWSTR,NTP(7),PLTIME(15),PRAT(20),PRFREQ,PSEXP,PSTAGTB(10),
     M TTPSTAG(10),MPSTAG,NPSTAG,PTMAX,PTMIN,QCTAB(10),TTABQC(10),MQC,
     N NQC,QRAT(20),
     O QRRAT(20),QRTAB(10),TTABQR(10),MQR,NQR,R(20),RIEXP,RNSI,RO,RODP,
     Q ROP,RS(20),RSSMAX,S,STEBOL,T(10,20),TAUC,TBTAB(10),TTABTB(10),
     S MTB,NTB,TDPRIME,THETA(20),TPRIME,XO,XORDER,ZS(20),ZSMAX
       REAL MDOTC,MDOT,MCDOT,MSDOT,MWSTR,MWO2,MACHNO,MDMAX
       INTEGER S,SM1,SM2
```

```
C
C STATION (1,1)     XI=0   ,  ETA=0
C
      DO 60 I=1,SM1
   60 CK(I)= (CKETA(I,1)+ CKETA(I+1,1))/2.0
      DELDE = DELTA(1)* DELETA
      PART2= H1(1,1) **2  * XDXISQ
      PART1=RODPC
      H1R = H1(1,1) * R(1)
      FF=CKXI(1,1)*(2.0-CORDSY)/(2.0*PART2)
      G=RO*CP(1,1)/DELTAU-2.0*PART1/H1R+8.0*PART1/(3.0*DELDE)
      H = 1.0/( H2(1,1)**2 * DELTA(1)**2)
      SC= H /(3.0* DELESQ)
      EPT4=SIGDP* (2.0/(H1R*H2(1,1)**2) - EIGHT3/DELDE)
      EPTB= EPT4 *TB
      EPT4= EPT4 *T(1,1)**3
      BSAVE  = G
      GO TO  (70, 80), IROCOL
   70 CONTINUE
      A(1,1) = 0.0
      BS1(1,1) = -SC*9.0 *CK(1)
      C(1,1)=  SC * (9.0 *CK(1) + CK(2) )
      Z(1) = -SC * CK(2)
      B(1)= BS1(1,1) - BSAVE + EPT4
      IF (IFIRST.EQ.0 )  GO TO 80
   78 DC(1) =(-BSAVE-BS1B(1,1))*T(1,1) - CB(1,1)*T(1,2)- ZB(1)* T(1,3)
     1 + EPTB
      GO TO 99
   80 FP=FF
      BS1B(1,1)= -7.0* FP
      CB(1,1)= 8.0 *FP
      ZB(1) = -FP
      IF (IFIRST.EQ.0 )  GO TO 78
   86 B(1) = BS1B(1,1)- BSAVE + EPT4
      DC(1) =(-BSAVE -BS1(1,1))*T(1,1) -C(1,1)*T(2,1) - Z(1)*T(3,1)
     1 + EPTB
   99 GO TO (101,600),IROCOL
C
C STATION(M,1)    , XI=0    ,  ETA LESS THAN 1 , GREATER THAN 0
C
      ENTRY COLXCM
  101 DO 200 M=N1,N2
      DELDE=DELTA(1)*DELETA
      MP1=M+1
      MM1 =M-1
      P817   =   8.0* DELTA(2) - DELTA(3) - 7.0* DELTA(1)
      PART2 = H1(M,1)**2   * XDXISQ
      CORD= (2.0-CORDSY)/2.0
      FF= CKXI(M,1)*CORD/PART2
      G = RO *CP(M,1)/DELTAU
      SC = 1.0 /(H2(M,1)* DELDE **2)
      H = FF* P817/(2.0* DELDE)   *ETA(M)
      P =  CKETA(M,1)/(H2(M,1)**2 *H1(M,1)* R(1) * DELDE)
      BSAVE =G
      GO TO (170,180), IROCOL
  170 CONTINUE
      U= ETA(M)*MDOT(1) * CP(M,1)/(2.0*DELTA(1) * DELETA)
      A(M,1)=  H -P + SC* CK(MM1) +U
      BS1(M,1) = SC * (-CK(MM1) - CK(M))
      C(M,1)= -H + P + SC* CK(M) -U
      B(M) = BS1(M,1) - BSAVE
      IF (IFIRST.EQ.0 )  GO TO 180
  178 DC(M)  =  (-BSAVE -BS1B(M,1))* T(M,1)-ZB(M)*T(M,3)-CB(M,1)*T(M,2)
      GO TO 200
  180 ZB(M) =   -FF
      CB(M,1)= 8.0 * FF
      BS1B(M,1)= -7.0*FF
      IF (IFIRST.EQ.0 )  GO TO 178
  190 B(1) = BS1B(M,1) - BSAVE
      DC(1)= (-BSAVE - BS1(M,1))*T(M,1)-A(M,1)*T(MM1,1)-C(M,1)*T(MP1,1)
  200 CONTINUE
      GO TO (202,600),IROCOL
```

```
C
C STATION (S,1)    ,XI=0    ,   ETA=1
C
      ENTRY COLXO1
  202 CORD=(2.0-CORDSY)/2.0
      FF=CKXI(S,1)*CORD/H1(S,1)**2
      DELDE=DELTA(1)*DELETA
      P =FF/XDXISQ
      H = 1.0/(H2(S,1)**2 *3.0* DELDE**2)
      G = RO*  CP(S,1)/ DELTAU
      SC = -9.0 * CK(SM1) * H
      BSAVE = G
      GO TO (270,280) ,IROCOL
  270 CONTINUE
      XX=CP(S,1)*MDOT(1)/(2.0*DELTA(1)*DELETA)
      V(1)= -CK(SM2)*H - XX
      A(S,1) = -SC + CK(SM2)*H + 4.0*XX
      DR=P*P817/CKETA(S,1)      *H2(S,1)
      DD = DR - 2.0/(H1(S,1)*R(1)*H2(S,1))-EIGHT3/
     1(H2(S,1) *DELDE)
      DDQS=DD*QS(1)
      BS1(S,1)=DD*SIG*T(S,1)**3 +SC-3.0*XX
      B(S) = BS1(S,1) -BSAVE
      IF (IFIRST.EQ.0 )  GO TO 280
  278 DC(S) = DDQS      +(-BSAVE - BS1B(S,1))*T(S,1)- CB(S,1)*T(S,2)
     1 - ZB(S) *T(S,3)
      GO TO 600
  280 CB(S,1)=8.0*P
      ZB(S) = -P
      BS1B(S,1) = -7.0*P
      IF (IFIRST.EQ.0 )  GO TO 278
  290 B(1) = BS1B(S,1) - BSAVE
      DC(1) =   (-BSAVE - BS1(S,1))*T(S,1)   - V(1)*
     1  T(SM2,1) - A(S,1) *T(SM1,1)   +DDQS
  600 RETURN
  201 FORMAT    (7E18.7)
      END
```

Subroutine COLMN.- Subroutine COLMN computes the coefficients of the tridiagonal matrix where $0 < \xi < 1$ and $0 \leqq \eta \leqq 1$. The flow chart for subroutine COLMN is as follows:






The program listing for subroutine COLMN is as follows:

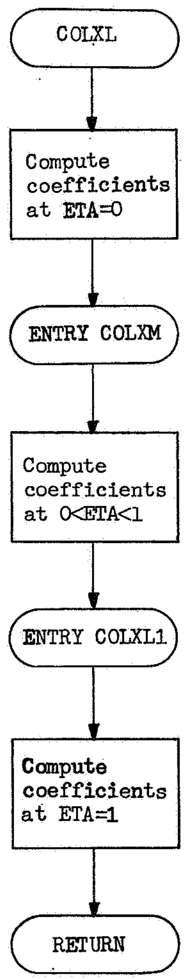
```
      SUBROUTINE COLMN (N1,N2,N)
C
C IROCOL = 1        COLUMN IMPLICIT
C IROCOL = 2        ROW   IMPLICIT
C
      COMMON /PICK/ A(10,20),AA(20),AB(10,20),ALPHA(20),B(20),
     2 BS1(10,20),BS1B(10,20),C(10,20),CB(10,20),CK(10),CKETA(10,20),
     4 CKXI(10,20),COST(20),CP(10,20),D(10,20),DC(20),
     6 DELESQ,DELETA,DELTA(20),DELXI,DELXISQ,E(10,20),EIGHT3,
     8 ELAM(20),ETA(10),EXPG,F(10,20),GG,GIMACH,H1(10,20),H2(10,20),
     A H3(10,20),HC(20),HCOMB(20),HE,HW(20),IFIRST,IROCOL,ITC,ITR,ITT,
     C ITTC,LM1,LM2,MCDOT(20),MDOT(22),MSDOT(20),PIO2,PRELOC(20),QC(20),
     E QC1,QCNET(20),QCOMB(20),QR(20),QR1,QS(20),RNS,RODPC,ROPCPP,
     G RSS(22),RSTO2,SIG,SIGDP,SIGMA,SIGP,SINT(20),SM1,SM2,TAU,TB,
     I TT(10,20),TTF(10,20),TWDELXI,TWOGI,V(20),VB(10),X(22),XODXISQ,
     K XODXI,Y(10,20),Z(20),ZB(10)
      COMMON /INPUTS/ DUMMY(1),AEXP,ALCTAB(10),TTALC(10),MALPHC,NALPHC,
     2 ALPHAT(10),TALPHA(10),MALPHA,NALPHA,ALSTAB(10),TTALS(10),MALPHS,
     4 NALPHS,ASEXP,BETA,BEXP,BSEXP,CE,CKETATB(50),TTCKETA(10),
     6 ETATAB(5),NCKETA,NETA,CKXITAB(50),TTCKXI(10),XITAB(5),NCKXI,
     8 NXI,CORDSY,CPDP,CPP,CPTAB(10),TTABCP(10),MCP,NCP,DELTAO(20),
     A DELTAU,DELTMIN,DTMAX,ELAMTB(28),TTELAM(7),PELAM(4),NELAM,
     C NPELAM,ENDTAU,EPSONE,EPSONEP,EPSONPP,ERRORT,GAMBAR,GAMINF,
     E HCOMBTB(28),TTHCOMB(7),PHCOMB(4),NHCOMB,NPHCOMB,HCTAB(28),
     G TTABHC(7),PHC(4),NHC,NPHC,HETAB(10),TTABHE(10),MHE,NHE,HWTAB(15),
     I TTABHW(15),MHW,NHW,IADJUST,IPLOT,L,MACHNO,MAXITT,MDMAX,
     J MDOTO(20),
     K MWO2,MWSTR,NTP(7),PLTIME(15),PRAT(20),PRFREQ,PSEXP,PSTAGTB(10),
     M TTPSTAG(10),MPSTAG,NPSTAG,PTMAX,PTMIN,QCTAB(10),TTABQC(10),MQC,
     N NQC,QRAT(20),
     O QRRAT(20),QRTAB(10),TTABQR(10),MQR,NQR,R(20),RIEXP,RNSI,RO,RODP,
     Q ROP,RS(20),RSSMAX,S,STEBOL,T(10,20),TAUC,TBTAB(10),TTABTB(10),
     S MTB,NTB,TDPRIME,THETA(20),TPRIME,XO,XORDER,ZS(20),ZSMAX
      REAL MDOTO,MDOT,MCDOT,MSDOT,MWSTR,MWO2,MACHNO,MDMAX
      DIMENSION DDQS(20),DDQSR(20)
      INTEGER S,SM1,SM2
C
C STATION (1,N)     XI GREATER THAN 0,LESS THAN 1     ETA=0
C
  201 FORMAT    (7E18.7)
      NM1 = N-1
      NP1 = N+1
      E32N=(H1(2,N)+H1(1,N))*(H3(2,N)+H3(1,N))*(CKETA(2,N)+CKETA(1,N))/
     1(4.*(H2(2,N)+H2(1,N)))
      E52N=(H1(3,N)+H1(2,N))*(H3(3,N)+H3(2,N))*(CKETA(3,N)+CKETA(2,N))/
     1(4.*(H2(3,N)+H2(2,N)))
      VV=1.0/ (3.0* DELTA(N)**2  * DELESQ )
      P1NP1=(H3(1,NP1)+H3(1,N))/(H1(1,NP1)+H1(1,N))
      P1NM1=(H3(1,NM1)+H3(1,N))/(H1(1,NM1)+H1(1,N))
```

```
      W=  H1(1,N)*H3(1,N)* DELETA *DELTA(N) *8.C
      G1N = H1(1,N)* H2(1,N) * H3(1,N) * RO *CP(1,N)
      YY=(-VV*W*RODPC-G1N/DELTAU)
      EPT4= -VV *W *SIGDP
      EPTB=  EPT4 * TB
      EPT4 = EPT4 * T(1,N)**3
      BSAVE = YY
      GC TO (170,180),IRCCOL
  170 CONTINUE
      BS1(1,N) = -VV* 9.0* E32N
      C(1,N)= VV *(9.0* E32N + E52N)
      Z(N) = -VV * E52N
      B(1)= BS1(1,N) + BSAVE +  EPT4
      IF (IFIRST.EQ.C )  GO TO 180
  178 DC(1) = (BSAVE - BS1B(1,N))*T(1,N) -AB(1,N)*T(1,NM1)-CB(1,N)*
     1 T(1,NP1) + EPTB
      GO TO 200
  180 D1NP1=(H2(1,NP1)+H2(1,N))*(H3(1,NP1)+H3(1,N))*(CKXI(1,NP1)+CKXI(1,
     1N))/(4.*XDXISQ*(H1(1,NP1)+H1(1,N)))
      D1NM1=(H2(1,NM1)+H2(1,N))*(H3(1,NM1)+H3(1,N))*(CKXI(1,NM1)+CKXI(1,
     1N))/(4.*XDXISQ*(H1(1,NM1)+H1(1,N)))
      AB(1,N)=D1NM1
      BS1B(1,N)=-D1NP1- D1NM1
      CB(1,N)=D1NP1
      IF (IFIRST.EQ.0 )  GO TO 178
  190 B(N)= BS1B(1,N) + BSAVE  + EPT4
      DC(N) = (BSAVE -BS1(1,N))*T(1,N) -C(1,N)*T(2,N) - Z(N)*T(3,N)
     1 + EPTB
  200 CONTINUE
      GC TO (202,800), IROCOL
C
C STATION (M,N)    XI GREATER THAN C, LESS THAN 1
C                 ETA GREATER THAN C, LESS THAN 1
C
      ENTRY COLMNMN
      NP1=N+1
      NM1=N-1
  202 DO 400 M=N1,N2
      MM1 = M-1
      MP1 = M+1
      VV= 1.0 /(DELTA(N)**2 * DELESQ)
      XX  =  ETA(M)*AA(N)/(DELTA(N)*  DELESQ)
      G   =  H1(M,N)* H2(M,N) * H3(M,N) *RO * CP(M,N)
      EMM12N=(H1(MM1,N)+H1(M,N))*(H3(MM1,N)+H3(M,N))*(CKETA(MM1,N)+CKETA
     1(M,N))/(4.*(H2(MM1,N)+H2(M,N)))*VV
      EMP12N=(H1(MP1,N)+H1(M,N))*(H3(MP1,N)+H3(M,N))*(CKETA(MP1,N)+CKETA
     1(M,N))/(4.*(H2(MP1,N)+H2(M,N)))*VV
      DMM12N=(H2(MM1,N)+H2(M,N))*(H3(MM1,N)+H3(M,N))*(CKXI(MM1,N)+CKXI(M
     1M1,N))/(4.*(H1(MM1,N)+H1(M,N)))
      DMP12N=(H2(MP1,N)+H2(M,N))*(H3(MP1,N)+H3(M,N))*(CKXI(MP1,N)+CKXI(M
     1P1,N))/(4.*(H1(MP1,N)+H1(M,N)))
      FMM12N=XX*DMM12N*AA(N)*(ETA(MM1)+ETA(M))/(DELTA(N)*2.)
      FMP12N=XX*DMP12N*AA(N)*(ETA(MP1)+ETA(M))/(DELTA(N)*2.)
      W = 4.0 * XC * DELXI * DELETA
      DENOM= 4.C* (DELTA(  NM1) + DELTA (  N)) *( H1(M,NM1) + H1 (M,N))
      FMNM12= (H3(M,NM1)+H3(M,N)) *(H2(M,NM1)+H2(M,N))* (CKXI(M,NM1)
     1+ CKXI(M,N))* (AA(  NM1) + AA(  N))* ETA(M)/DENOM
      DENOM= 4.C* (DELTA(  NP1)+DELTA(  N))*(H1(M,NP1)+H1(M,N))
      FMNP12= (H3(M,NP1)+H3(M,N))*(H2(M,NP1) +H2(M,N))*(CKXI(M,NP1)
     1 +CKXI(M,N))*(AA(  NP1)+AA(  N))*ETA(M)/DENOM
      D1 = (FMNP12*(T(MP1,NP1)-T(MM1,NP1)+T(MP1,N)-T(MM1,N))-FMNM12*
     1 (T(MP1,N)-T(MM1,N)+T(MP1,NM1)-T(MM1,NM1)))/W
      D2 = ETA(M) *AA(N)* CKXI(M,N)* (H2(MP1,N)* H3(MP1,N)* (T(MP1,NP1)
     1 - T(MP1,NM1))/H1(MP1,N)    - H2(MM1,N) * H3(MM1,N)* (T(MM1,NP1)
     2 - T(MM1,NM1))/H1(MM1,N) ) /(DELTA(N) * W)
      DS = D1  + D2  - G *T(M,N )/ DELTAU
      BSAVE = G/DELTAU
      GO TO  (370,380),IROCOL
```

```
 370 CONTINUE
     HMN = ETA(M) * MDOT(N)/(DELTA(N) * RO)
     YY=  G * HMN/(2.0* DELETA)
     A(M,N) = EMM12N + FMM12N + YY
     BS1(M,N) = -EMM12N - EMP12N -FMP12N - FMM12N
     C(M,N) = EMP12N + FMP12N - YY
     B(M) = BS1(M,N) - BSAVE
     IF (IFIRST.EQ.0 )  GO TO 380
 378 DC(M) = DS- BS1B(M,N)* T (M,N) -AB(M,N)*T(M,NM1)-CB(M,N)*T(M,NP1)
     GO TO 400
 380 DMNM12=(H2(M,NM1)+H2(M,N))*(H3(M,NM1)+H3(M,N))*(CKXI(M,NM1)+CKXI(M
    1,N))/(4.*(H1(M,NM1)+H1(M,N)))
     AB(M,N)=DMNM12/XDXISQ
     DMNP12=(H2(M,NP1)+H2(M,N))*(H3(M,NP1)+H3(M,N))*(CKXI(M,NP1)+CKXI(M
    1,N))/(4.*(H1(M,NP1)+H1(M,N)))
     CB(M,N)=DMNP12/XDXISQ
     BS1B(M,N)= -AB(M,N) - CB(M,N)
     IF (IFIRST.EQ.0 )  GO TO 378
 390 B(N) = BS1B(M,N) - BSAVE
     DC(N) = DS - BS1(M,N)*T(M,N) - A(M,N)*T(MM1,N)- C(M,N)*T(MP1,N)
 400 CONTINUE
     GO TO  (401,800), IROCOL
C
C STATION (S,N)    XI GREATER THAN 0, LESS THAN 1 ,     ETA =1
C
     ENTRY  COLMN1
     NP1=N+1
     NM1=N-1
 401 H1H3 = H1(S,N)* H3(S,N)
     XX= 3.0 * DELTA(N)**2 * DELESQ
     U = AA(N)/ (3.0                            *DELESQ* DELTA(N)    )
     G =             H1H3  *H2(S,N) * RO *CP(S,N)
     PART=AA(N)/(DELTA(N)*4.0*DELETA*DELXI*XO)
     SST=      H3(S,N)*CKXI(S,N)*3./H1(S,N)
     DS=PART*(SST*T(S,NP1)-SST*T(S,NM1)
    1     -4.0*H2(SM1,N)*H3(SM1,N)*CKXI(SM1,N)*(T(SM1,NP1)-T(SM1,NM1))/
    2H1(SM1,N)+H2(SM2,N)*H3(SM2,N)*CKXI(SM2,N)* (T(SM2,NP1)-T(SM2,NM1))
    3/H1(SM2,N))
     ESM32N=(H1(SM1,N)+H1(SM2,N))*(H3(SM1,N)+H3(SM2,N))*(CKETA(SM1,N)+
    1CKETA(SM2,N))/(4.*(H2(SM1,N)+H2(SM2,N))*XX)
     ESM12N=(H1(SM1,N)+H1(S,N))*(H3(SM1,N)+H3(S,N))*(CKETA(SM1,N)+CKETA
    1(S,N))/(4.*(H2(SM1,N)+H2(S,N))*XX)*9.
     DSM12N=(H2(SM1,N)+H2(S,N))*(H3(SM1,N)+H3(S,N))*(CKXI(SM1,N)+CKXI(S
    1 ,N))/ (4.0*(H1(SM1,N)+ H1(S,N)))
     FSM12N=DSM12N*AA(N)*(ETA(SM1)+ETA(S))/(DELTA(N)*2.)*9.*U
     DSM32N=(H2(SM2,N)+H2(SM1,N))*(H3(SM2,N)+H3(SM1,N))*(CKXI(SM2,N)+
    1 CKXI(SM1,N))/(4.*(H1(SM2,N)+H1(SM1,N)))
     FSM32N=DSM32N*AA(N)*(ETA(SM2)+ETA(SM1))/(DELTA(N)*2.)*U
     BSAVE = G/DELTAU
     GO TO  (570,580),IROCOL
 570 CONTINUE
     YY=G*MDOT(N)/ (RO*2.0*DELTA(N)*DELETA)
     V(N)= -ESM32N - FSM32N -YY
     A(S,N) = ESM12N + ESM32N + FSM12N + FSM32N + 4.0*YY
     DD=8.*H1H3*DELTA(N)*DELETA/XX  + 8.*U*
    1H2(S,N)*DELTA(N)*F(S,N)*DELETA/CKETA(S,N)
     DDQS(N)=DD*QS(N)
     BS1(S,N)=-DD*SIG*T(S,N)**3-ESM12N-FSM12N-3.0*YY
     B(S) = BS1(S,N) - BSAVE
     IF (IFIRST.EQ.0 )  GO TO 580
 578 DC(S) =-DD  QS(N)+ DS+(-BSAVE-BS1B(S,N))*T(S,N)-AB(S,N)*T(S,NM1)
    1 -CB(S,N)*T(S,NP1)+ DDQSR(N)
     GO TO 600
 580 DSNM12=(H2(S,NM1)+H2(S,N))*(H3(S,NM1)+H3(S,N))*(CKXI(S,NM1)+CKXI(S
    1,N))/(4.*(H1(S,NM1)+H1(S,N)) *XDXISQ)
     DSNP12=(H2(S,NP1)+H2(S,N))*(H3(S,NP1)+H3(S,N))*(CKXI(S,NP1)+CKXI(S
    1,N))/(4.*(H1(S,NP1)+H1(S,N))  *XDXISQ)
     DENOM=4.0*(DELTA(   NM1)+DELTA(   N))*(H2(S,NM1)+H1(S,N))
     FSNM12= (H3(S,NM1)+H3(S,N))*(H2(S,NM1)+H2(S,N))* (CKXI(S,NM1)
```

```
      1 +CKXI(S,N))*(AA(   NM1)+AA(   N))/DENCM
       DENCM=4.0*(DELTA(   NP1)+DELTA(   N))*(H1(S,NP1)+H1(S,N))
       FSNP12= (H3(S,NP1)+H3(S,N))*(H2(S,NP1)+H2(S,N))*(CKXI(S,NP1)
      1 +CKXI(S,N))*(AA(   NP1)+AA(   N))/DENOM
       DENOM=2.0*XC*DELXI
       QSN= DELTA(N)*H2(S,N)/(CKETA(S,N)*DENCM)
       QSNP1= DELTA(NP1)* H2(S,NP1)/(CKETA(S,NP1)* DENCM)
       QSNM1= DELTA(NM1)* H2(S,NM1)/(CKETA(S,NM1)* DENOM)
       DDQSR(N)= FSNP12* (QSNP1*QS(NP1)+ QSN*QS(N))-FSNM12*
      1(QSN*QS(N)+ QSNM1*QS(NM1))
       AB(S,N)=DSNM12-FSNM12*SIG*QSNM1*T(S,NM1)**3
       CB(S,N)=DSNP12+FSNP12*QSNP1*SIG*T(S,NP1)**3
       BS1B(S,N)=-DSNP12-DSNM12+SIG*T(S,N)**3*QSN *(FSNP12-FSNM12)
       IF (IFIRST.EQ.0 )  GO TO 578
  590  B(N)=BS1B(S,N)-BSAVE
       DC(N)  =  -CD QS(N)  +(-BSAVE-BS1(S,N))*T(S,N)+DS
      1-A(S,N)*T(SM1,N)-V(N)*T(SM2,N)+ DDQSR(N)
  600  CONTINUE
  800  RETURN
       END
```

Subroutine COLXL.- Subroutine COLXL computes the coefficients of the tridiagonal matrix where $\xi = 1$ and $0 \leqq \eta \leqq 1$. The flow chart for subroutine COLXL is as follows:



The program listing for subroutine COLXL is as follows:

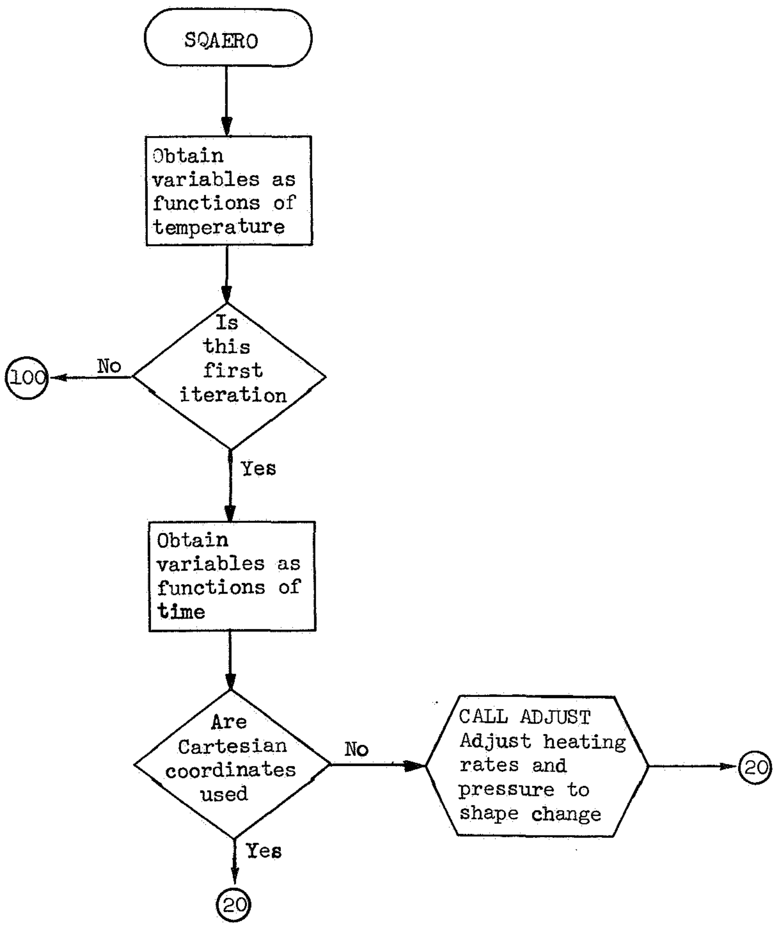
```
      SUBROUTINE COLXL(N1,N2)
C
C COMPUTES COEF. FOR  XI=1 ( X=L)   COLUMN IMPLICIT
C IROCOL = 1         COLUMN IMPLICIT
C IROCOL = 2         ROW   IMPLICIT
C
      COMMON /PICK/ A(10,20),AA(20),AB(10,20),ALPHA(20),B(20),
     2 BS1(10,20),BS1B(10,20),C(10,20),CB(10,20),CK(10),CKETA(10,20),
     4 CKXI(10,20),CCST(20),CP(10,20),D(10,20),DC(20),
     6 DELESQ,DELETA,DELTA(20),DELXI,DELXISQ,E(10,20),EIGHT3,
     8 ELAM(20),ETA(10),EXPG,F(10,20),GG,GIMACH,H1(10,20),H2(10,20),
     A H3(10,20),HC(20),HCOMB(20),HE,HW(20),IFIRST,IROCOL,ITC,ITR,ITT,
     C ITTC,LM1,LM2,MCDOT(20),MDOT(22),MSDOT(20),PID2,PPELCC(20),QC(20),
     E QC1,QCNET(20),QCCMB(20),QR(20),QR1,QS(20),RNS,RODPC,ROPCPP,
     G RSS(22),RST02,SIG,SIGDP,SIGMA,SIGP,SINT(20),SM1,SM2,TAU,TB,
     I TT(10,20),TTF(10,20),TWDELXI,TWOGI,V(20),VB(10),X(22),XDXISQ,
     K XCDXI,Y(10,20),Z(20),ZB(10)
      COMMON /INPUTS/ DUMMY(1),AEXP,ALCTAB(10),TTALC(10),MALPHC,NALPHC,
     2 ALPHAT(10),TALPHA(10),MALPHA,NALPHA,ALSTAB(10),TTALS(10),MALPHS,
     4 NALPHS,ASEXP,BETA,BEXP,BSEXP,CE,CKETATB(50),TTCKETA(10),
     6 ETATAB(5),NCKETA,NETA,CKXITAB(50),TTCKXI(10),XITAB(5),NCKXI,
     8 NXI,CORDSY,CPDP,CPP,CPTAB(10),TTABCP(10),MCP,NCP,DELTAO(20),
     A DELTAU,DELTMIN,DTMAX,ELAMTB(28),TTELAM(7),PELAM(4),NELAM,
     C NPELAM,ENDTAU,EPSONE,EPSCNEP,EPSCNPP,ERRORT,GAMBAR,GAMINF,
     E HCCMBTB(28),TTHCOMB(7),PHCOMB(4),NHCOMB,NPHCOMB,HCTAB(28),
     G TTABHC(7),PHC(4),NHC,NPHC,HETAB(10),TTABHE(10),MHE,NHE,HWTAB(15),
     I TTABHW(15),MHW,NHW,IADJUST,IPLOT,L,MACHNO,MAXITT,MDMAX,
     J MDOTO(20),
     K MWO2,MWSTR,NTP(7),PLTIME(15),PRAT(20),PRFREQ,PSEXP,PSTAGTB(10),
     M TTPSTAG(10),MPSTAG,NPSTAG,PTMAX,PTMIN,QCTAB(10),TTABQC(10),MQC,
     N NQC,QRAT(20),
     O QRRAT(20),QRTAB(10),TTABQR(10),MQR,NQR,R(20),RIEXP,RNSI,RO,RODP,
     Q ROP,RS(20),RSSMAX,S,STEBOL,T(10,20),TAUO,TBTAB(10),TTABTB(10),
     S MTB,NTB,TDPRIME,THETA(20),TPRIME,XO,XORDER,ZS(20),ZSMAX
      REAL MDOTC,MDOT,MCDOT,MSDOT,MWSTR,MWO2,MACHNO,MDMAX
      DIMENSION AL(10)
      INTEGER S,SM1,SM2
  201 FORMAT    (7E18.7)
C
C STATION  (1,L)      X= L   ,    ETA =0,
C
      W= 8.0* XO**2 * DELXI
      U= 8.0*H2(1,L)*H3(1,L) *XO
      XX = 8.0 * H1(1,L) * H3(1,L)* DELTA(L)
      SC=  8.0*  DELTA(L)**2  * DELETA
      G= -U*ROPCPP/W-XX*RODPC/SC - H1(1,L)*H2(1,L)*H3(1,L)*RO*CP(1,L)
     1 /DELTAU
      PART1 =    SC * DELETA
      E32L=(H1(2,L)+H1(1,L))*(H3(2,L)+H3(1,L))*(CKETA(2,L)+CKETA(1,L))/
     1(4.*(H2(2,L)+H2(1,L)))*9.
      E52L=(H1(3,L)+H1(2,L))*(H3(3,L)+H3(2,L))*(CKETA(3,L)+CKETA(2,L))/
     1(4.*(H2(3,L)+H2(2,L)))
      D1LM32=(H2(1,LM1)+H2(1,LM2))*(H3(1,LM1)+H3(1,LM2))*(CKXI(1,LM1)+
     1CKXI(1,LM2))/(4.*(H1(1,LM1)+H1(1,LM2)))
      D1LM12=(H2(1,LM1)+H2(1,L))*(H3(1,LM1)+H3(1,L))*(CKXI(1,LM1)+CKXI(1
     1,L))/(4.*(H1(1,LM1)+H1(1,L)))
      EPT4=  (-U*SIGP/W  -XX *SIGDP/SC)
      EPTB= EPT4* TB
      EPT4= EPT4* T(1,L) **3
      BSAVE = G
C
      GO TO ( 150,180),IROCOL
  150 CONTINUE
      BS1(1,L)= -E32L/PART1
      C(1,L)= (E52L + E32L)/PART1
      Z(L)= -E52L/PART1
      B(1)= BS1(1,L) + BSAVE  + EPT4
      IF (IFIRST.EQ.0 )  GO TO 180
```

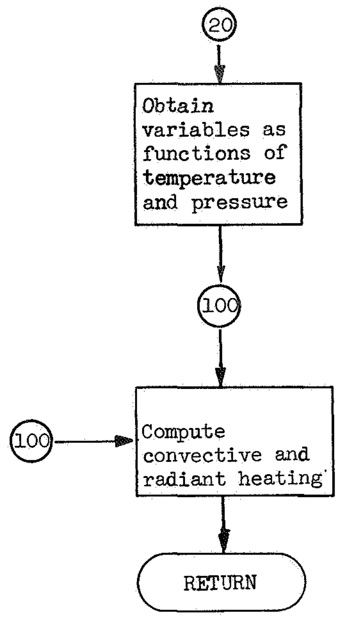
```
 178 DC(1) = (BSAVE - BS1B(1,L))* T(1,L) -VB(1)*T(1,LM2)- AB(1,L)*
    1  T(1,LM1) + EPTB
     GO TO 198
 180 CONTINUE
     VB(1)=- D1LM32/(W*DELXI)
     AB(1,L)= (D1LM32+ 9.0*D1LM12)/(W*DELXI)
     BS1B(1,L)=-9.0*D1LM12/ (W*DELXI)
     IF (IFIRST.EQ.0 )  GO TO 178
 190 B(L) = BS1B(1,L) + BSAVE  + EPT4
     DC(L) = (BSAVE -BS1(1,L))*T(1,L)- C(1,L)*T(2,L) -Z(L)*T(3,L)
    1 +EPTB
 198 CONTINUE
     GO TO (202,800),IROCOL
C
C STATION (M,L)     X=L       ETA GREATER THAN 0, LESS THAN 1
C
     ENTRY  COLXLM
 202 DO 210 M=1,S
 210 AL(M) =  H2(M,L)*H3(M,L)/H1(M,L)
     W= 3.0 * XC * DELXI
     YY = DELTA(L) **2  * DELESQ
     DO 300 M=N1,N2
     MM1 = M-1
     MP1 = M+1
     XX = ETA(M) *(AA(L)+ AA(LM1))/(4.0* (DELTA(L)+ DELTA(LM1))*DELETA)
     XX1= ETA(M) * (AA(LM1)+AA(LM2))/(4.*(DELTA(LM1)+DELTA(LM2)) *
    1DELETA)
     XY = 8.0* C(M,L) * H1(M,L)/ CKXI(M,L)
     AN = ETA(M) *AA(L)* CKXI(M,L)/ DELTA(L)
     AM = AN/(DELTA(L) * DELESQ)
     G = H1(M,L)* H2(M,L)* H3(M,L) * RO * CP(M,L)
     AJ =  AN /(4.0* DELETA * XO  * DELXI )
     U1= (H2(MP1,L)+H2(M,L))* (H3(MP1,L)+H3(M,L)) *(ETA(MP1)+ETA(M))
    1 /(4.0* (H1(MP1,L)+H1(M,L)))*AA(L)
     U2= (H2(MM1,L)+H2(M,L)) * (H3(MM1,L)+H3(M,L)) *(ETA(MM1)+ETA(M))
    1/ (4.0* (H1(MM1,L) +H1(M,L)))*AA(L)
     DMLM32=(H2(M,LM1)+H2(M,LM2))*(H3(M,LM1)+H3(M,LM2))*(CKXI(M,LM1)+
    1CKXI(M,LM2))/(4.*(H1(M,LM1)+H1(M,LM2)))
     DMLM12=(H2(M,LM1)+H2(M,L))*(H3(M,LM1)+H3(M,L))*(CKXI(M,LM1)+CKXI(M
    1,L))/(4.*(H1(M,LM1)+H1(M,L)))
     D1= -9.0*DMLM12* XX* (T(MP1,L)-T(MM1,L)+
    1 T(MP1,LM1) - T(MM1,LM1))
     D2 = DMLM32 *(-XX1)* (T(MP1,LM1)-T(MM1,LM1)
    1 + T(MP1,LM2)- T(MM1,LM2))
     DN =- (D1 +D2)/W
     DN1 = AJ *( AL(MP1)* (3.0*T(MP1,L)-4.0*T(MP1,LM1)+T(MP1,LM2))
    1- AL(MM1   ) *(3.0*T(MM1,L)- 4.0*T(MM1,LM1) + T(MM1,LM2))  )
     BSAVE = -RCPCPP * XY/ W  - G/DELTAU
     EPT4=  -SIGP * XY /W
     EPTB= EPT4 * TB
     EPT4= EPT4 * T(M,L) **3
C
     GO TO ( 240, 280),IROCOL
 240 CONTINUE
     EMM12 = (H1(M,L)+H1(MM1,L)) *(H3(M,L)+H3(MM1,L)) *(CKETA(M,L)
    1 +CKETA(MM1,L))/ (4.0* (H2(M,L)+H2(MM1,L)))
     EMP12= (H1(M,L)+H1(MP1,L)) *(H3(M,L)+H3(MP1,L)) *( CKETA(M,L)
    1+ CKETA(MP1,L))/ (4.0* (H2(M,L) +H2(MP1,L)))
     GH =G * ETA(M)* MDOT(L)/(DELTA(L)*RO *2.0* DELETA)
     A(M,L)= AM*U2 + EMM12/YY +GH
     C(M,L)= AM*U1 + EMP12/YY -GH
     BS1 (M,L)= AM* (-U1-U2) + (-EMM12 -EMP12)/YY
     B(M) = BS1(M,L) + BSAVE  + EPT4
     IF (IFIRST.EQ.0 )  GO TO 280
 278 DC(M) = DN + DN1 + BSAVE*T(M,L) - VB(M   )*T(M,LM2) -AB(M,L)*
    1 T(M,LM1)- BS1B(M,L)* T(M,L) + EPTB
     GO TO 300
```

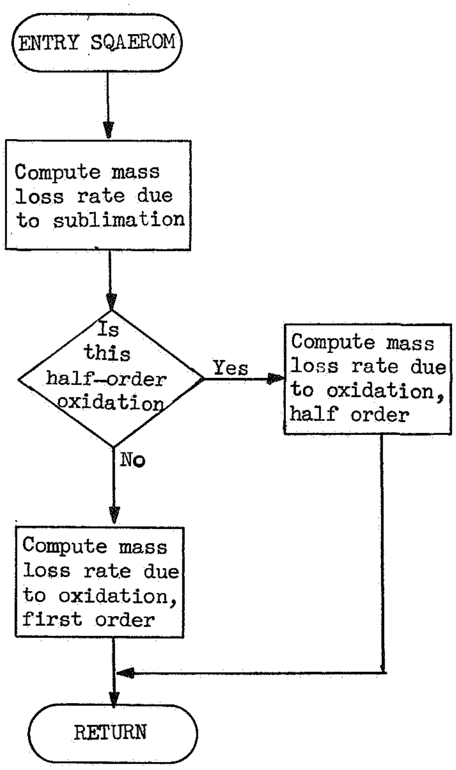
```
  280 CONTINUE
      PART =W * XC * DELXI
      PART2=DMLM32/PART
      PART1= 9.0 * DMLM12/PART
      VB(M)     = - PART2
      AB(M,L)   = PART1   + PART2
      BS1B(M,L) =- PART1
      IF (IFIRST.EQ.0 )  GO TO 278
  290 B(L)=  BS1B(M,L) + BSAVE + EPT4
      DC(L)=DN+DN1+(BSAVE-BS1(M,L))*  T(M,L)- A(M,L)*T(MM1,L)- C(M,L)
     1 * T(MP1,L) + EPTB
  300 CONTINUE
      GO TO  (301,800),IROCOL
C
C STATION (S,L)    XI =1, (X=L)  ,   ETA=1,
C
      ENTRY COLXL1
  301 CONTINUE
      W = 3.0 * XODXI
      WSQ = 3.0* XDXISQ
      DEDETA =  DELTA(L) * DELETA
      TWDEL =  2.0* DELTA(L)
      U1=(AA(L)+AA(LM1))/(2.*(DELTA(L)+DELTA(LM1)))
      U2=(AA(LM1)+AA(LM2))/(2.*(DELTA(LM1)+DELTA(LM2)))
      SP=(H1(S,L)* XODXI+ 2.0*TPRIME)/(H1(S,L)*XODXI)
      DHK = DELTA(L) * H2(S,L)/ CKETA(S,L)    *SP
      DHK1=  DELTA(LM1)* H2(S,LM1)/ CKETA(S,LM1)
      DHK2=  DELTA(LM2)* H2(S,LM2)/ CKETA(S,LM2)
      ZZ2 =8.0* DELETA * E(S,L)* DELTA(L) * H2(S,L)* SP/CKETA(S,L)
      FF=1.0/(3.0*DEDETA**2)
      H = 8.0 * H1(S,L) * D(S,L)/CKXI(S,L)
      PART = AA(L) /DEDETA
      ADD = PART/3.0
      ADD1 = (1.0 + ETA(SM1)) * PART/2.0
      ADD2 =    (ETA(SM1) + ETA(SM2)) *PART/2.0
      PART =       3.0* T(SM1,L)-4.0*T(SM1,LM1)+ T(SM1,LM2)
      DSM32L=(H2(SM2,L)+H2(SM1,L))*(H3(SM2,L)+H3(SM1,L))*(CKXI(SM2,L)+
     1 CKXI(SM1,L))/(4.*(H1(SM2,L)+H1(SM1,L)))
      PART2=DSM32L*(3.*T(SM2,L)-4.*T(SM2,LM1)+T(SM2,LM2)+PART)
      DSM12L= (H2(SM1,L)+H2(S,L))* (H3(SM1,L)+H3(S,L))* (CKXI(SM1,L)
     1 +CKXI(S,L))/(4.0* (H1(SM1,L)+H1(S,L)))
      PART1= -9.0*DSM12L*(3.0*T(S,L)-4.0*T(S,LM1)+T(S,LM2)+PART)
      GSL =  H1(S,L)* H2(S,L)*H3(S,L)* RO *CP(S,L)
      PARTW = -1.0/W + ADD
      EPT4= SIGP * H*PARTW
      EPTB = EPT4 * TB
      EPT4 = EPT4 * T(S,L) **3
      DN = ADD * (PART1 + PART2)/(4.0* XODXI)  + EPTB
      BSAVE=H*RCPCPP*(PARTW)-GSL/DELTAU
      GO TO (550,650),IROCOL
  550 CONTINUE
      AJ=GSL *MDOT(L)/ (RO*2.0*DELTA(L)*DELETA)
      DDSL= -FF*ZZ2
      QSAVE= DDSL* QS(L)
      ESM32L=(H1(SM2,L)+H1(SM1,L))* (H3(SM2,L)+H3(SM1,L))* (CKETA(SM2,L)
     1+ CKETA(SM1,L))/ (4.0*(H2(SM2,L)+H2(SM1,L)))
      PARTE3=FF*ESM32L
      PARTD3= ADD*ADD2*DSM32L
      V(L)= -PARTD3- PARTE3- AJ
      ESM12L= (H1(SM1,L)+H1(S,L))*(H3(SM1,L)+H3(S,L))*(CKETA(SM1,L)
     1 +CKETA(S,L))/(4.0* (H2(SM1,L)+H2(S,L)))
      PARTE1 = FF*9.0*ESM12L
      PARTD1= ADD*ADD1*9.0*DSM12L
      A(S,L)= PARTD1 + PARTD3 + PARTE3 + PARTE1 + 4.0*AJ
      BS1(S,L)= DDSL*SIG*T(S,L)**3 - PARTD1 - PARTE1 -3.0*AJ
      B(S)=  BS1(S,L) +   BSAVE + EPT4
      IF (IFIRST.EQ.0 )  GO TO 650
  648 DC( S) =  DN - VB(S  ) *T(S,LM2) -AB(S,L)*T(S,LM1)- (BS1B(S,L)
     1 -BSAVE) * T(S,L)+ QSAVE + DDQSR
      GO TO 300
```

```
650 CONTINUE
      WXODXI = W* XODXI
      DSLM3=(H2(S,LM1)+H2(S,LM2))*(H3(S,LM1)+H3(S,LM2))* (CKXI(S,LM1)
     1+CKXI(S,LM2))/(4.0* (H1(S,LM1)+H1(S,LM2)))
      DSLM1=9.0* (H2(S,L)+H2(S,LM1)) *(H3(S,L)+H3(S,LM1)) *(CKXI(S,L)+
     1 CKXI (S,LM1))/ (4.0*(H1(S,L)+H1(S,LM1)))
      QSLM1 = (-DSLM1 *U1 + DSLM3* U2) *DHK1/W
      QSLM2 = DSLM3*U2 *DHK2/W
      QSL=-U1* DHK* DSLM1/W
      DDQSR= QSLM1* QS(LM1) +QSLM2 *QS(LM2) + QSL*QS(L)
      VB(S)=-DSLM3/WXODXI+QSLM2*SIG*T(S,LM2)**3
      AB(S,L)=(DSLM1+DSLM3)/WXODXI+QSLM1*SIG*T(S,LM1)**3
      BS1B(S,L)=-DSLM1/WXODXI+QSL*SIG*T(S,L)**3
      IF (IFIRST.EQ.0 )  GO TO 648
690   B(L) = BS1B(S,L) + BSAVE + EPT4
      DC(L)=DN+QSAVE + DDQSR
     1 - V(  L) *T(SM2,L) - A(S,L) *T(SM1,L) - (BS1(S,L)-BSAVE)*T(S,L)
800 RETURN
      END
```

Subroutine SQAERO. - Subroutine SQAERO computes convective and radiant heating rates and surface mass-loss rates and obtains variables which are functions of time, temperature, and pressure. The flow chart for subroutine SQAERO is as follows:




20
Obtain variables as functions of temperature and pressure
100
100
Compute convective and radiant heating
RETURN

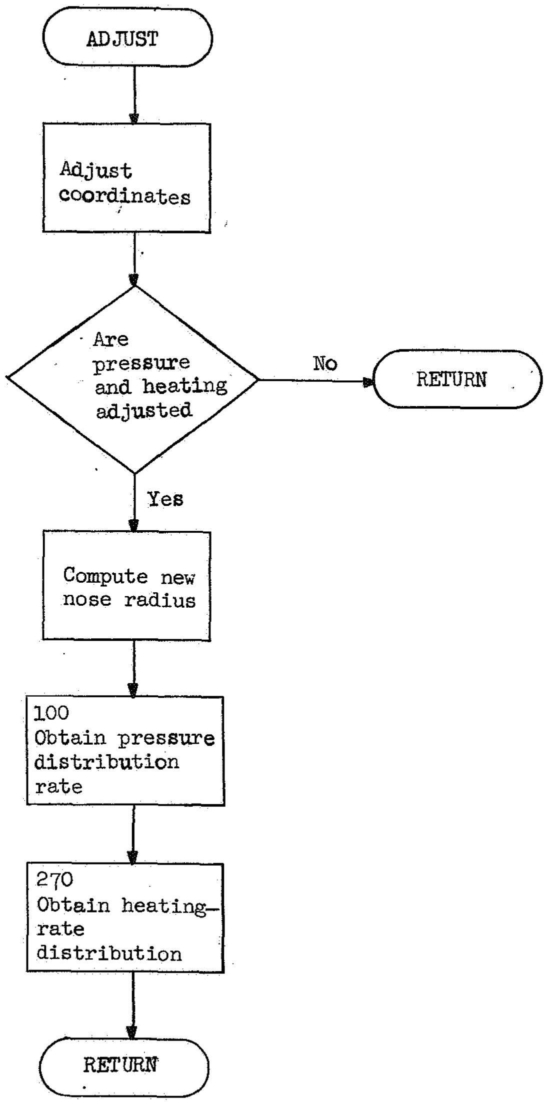

ENTRY SQAEROM
Compute mass loss rate due to sublimation
Is this half-order oxidation
Yes
No
Compute mass loss rate due to oxidation, half order
Compute mass loss rate due to oxidation, first order
RETURN

The program listing for subroutine SQAERO is as follows:

```
      SUBROUTINE SQAERO
C
C  THIS ROUTINE COMPUTES THE HEATING RATES AND THE MASS LOSS RATES
C
      COMMON /PICK/ A(10,20),AA(20),AB(10,20),ALPHA(20),B(20),
     2 BS1(10,20),BS1B(10,20),C(10,20),CB(10,20),CK(10),CKETA(10,20),
     4 CKXI(10,20),COST(20),CP(10,20),D(10,20),DC(20),
     6 DELESQ,DELETA,DELTA(20),DELXI,DELXISQ,E(10,20),EIGHT3,
     8 ELAM(20),ETA(10),EXPG,F(10,20),GG,GIMACH,H1(10,20),H2(10,20),
     A H3(10,20),HC(20),HCOMB(20),HE,HW(20),IFIRST,IROCCL,ITC,ITR,ITT,
     C ITTO,LM1,LM2,MCDOT(20),MDOT(22),MSDOT(20),PID2,PRELOC(20),QC(20),
     E QC1,QCNET(20),QCCMB(20),QR(20),QR1,QS(20),RNS,RODPC,ROPCPP,
     G RSS(22),RSTO2,SIG,SIGDP,SIGMA,SIGP,SINT(20),SM1,SM2,TAU,TB,
     I TT(10,20),TTF(10,20),TWDELXI,TWOGI,V(20),VB(10),X(22),XDXISQ,
     K XDDXI,Y(10,20),Z(20),ZB(10)
      COMMON /INPUTS/ DUMMY(1),AEXP,ALCTAB(10),TTALC(10),MALPHC,NALPHC,
     2 ALPHAT(10),TALPHA(10),MALPHA,NALPHA,ALSTAB(10),TTALS(10),MALPHS,
     4 NALPHS,ASEXP,BETA,BEXP,BSEXP,CE,CKETATB(50),TTCKETA(10),
     6 ETATAB(5),NCKETA,NETA,CKXITAB(50),TTCKXI(10),XITAB(5),NCKXI,
     8 NXI,CORDSY,CPCP,CPP,CPTAB(10),TTABCP(10),MCP,NCP,DELTAO(20),
     A DELTAU,DELTMIN,DTMAX,ELAMTB(28),TTELAM(7),PELAM(4),NELAM,
     C NPELAM,ENDTAU,EPSONE,EPSONEP,EPSONPP,ERRORT,GAMBAR,GAMINF,
     E HCOMBTB(28),TTHCCMB(7),PHCOMB(4),NHCOMB,NPHCCMB,HCTAB(28),
     G TTABHC(7),PHC(4),NHC,NPHC,HETAB(10),TTABHE(10),MHE,NHE,HWTAB(15),
     I TTABHW(15),MHW,NHW,IADJUST,IPLOT,L,MACHNO,MAXITT,MDMAX,
     J MDOTO(20),
     K MWC2,MWSTR,NTP(7),PLTIME(15),PRAT(20),PRFREQ,PSEXP,PSTAGTB(10),
     M TTPSTAG(10),MPSTAG,NPSTAG,PTMAX,PTMIN,QCTAB(10),TTABQC(10),MQC,
     N NQC,QRAT(20),
     O QRRAT(20),QRTAB(10),TTABQR(10),MQR,NQR,R(20),RIEXP,RNSI,RO,RODP,
     Q RCP,RS(20),RSSMAX,S,STEBOL,T(10,20),TAUO,TBTAB(10),TTABTB(10),
     S MTB,NTB,TDPRIME,THETA(20),TPRIME,XO,XCRDER,ZS(20),ZSMAX
      REAL MDOTC,MDOT,MCDOT,MSDOT,MWSTR,MWO2,MACHNO,MDMAX
      INTEGER S,SM1,SM2
C LOOK UP  CP,  CPBAR,  CKN  ,ETC.  AS  FUNCTIONS  OF  TEMPERATURE
      DO 11 N=1,L
      CALL FTLUP (TT(S,N),ALPHA(N),MALPHA,NALPHA,TALPHA,ALPHAT)
   11 CALL FTLUP (TT(S,N),HW(N),MHW,NHW,TTABHW,HWTAB)
      IF (ITT.NE.1) GO TO 100
C LOOK UP  FUNCTIONS  OF  TIME
      CALL FTLUP (TAU,ALPHAC,MALPHC,NALPHC,TTALC,ALCTAB)
      CALL FTLUP (TAU,ALPHAS,MALPHS,NALPHS,TTALS,ALSTAB)
      CALL FTLUP (TAU,HE,MHE,NHE,TTABHE,HETAB)
      CALL FTLUP (TAU,PSTAG,MPSTAG,NPSTAG,TTPSTAG,PSTAGTB)
      CALL FTLUP (TAU,QC1   ,MQC,NQC,TTABQC,QCTAB)
      CALL FTLUP (TAU,QR1,MQR,NQR,TTABQR,QRTAB)
      CALL FTLUP (TAU,TB,MTB,NTB,TTABTB,TBTAB)
      TB =TB**4
C
C ADJUST CONVECTIVE AND RADIANT HEATING RATES AND THE PRESSURE AND
C HEATING DISTRIBUTION TO SHAPE CHANGE  (ADJUST  QC1,QR1,PRAT,QRAT )
C
      IF (CORDSY.NE.0) GO TO 20
      CALL ADJUST
   20 DO 30 N=1,L
      DELTAO(N)=DELTA(N)
      QR(N) = QR1 * QRRAT(N)
      QC(N)= QC1 *QRAT(N)
      PRELOC(N) = PSTAG * PRAT(N)
      CALL DISCCT( TT(S,N),PRELOC(N),TTABHC,HCTAB,PHC,11,28,4,HC(N))
      CALL DISCCT(TT(S,N),PRELOC(N),TTELAM,ELAMTB,PELAM,11,28,4,ELAM(N)
     1)
   30 CALL DISCCT (TT(S,N),PRELOC(N),TTHCCMB,HCOMBTB,PHCOMB,11,28,4,
     1 HCOMB(N))
```

```
C COMPUTE QS ACROSS FRONT SURFACE
      BAT = 1.0 - BETA
  100 DO 200 N=1,L
      CELL =HE /QC(N)
      CAT = QC(N) * (1.C - HW(N)/HE)
      BLOCK=(ALPHAC *MCDOT(N) + ALPHAS *MSDOT(N))* CELL
      QCNET(N) =  CAT *(1.0 - BAT *(0.6* BLOCK - C.084 * BLOCK**2)
     1- BETA * BLOCK)
      QCOMB(N)= MCDOT(N) * HCOMB(N)
      QS(N)= QCNET(N) + ALPHA* QR(N)- MSDOT(N)*HC(N)+ QCOMB(N)
  200 CONTINUE
      RETURN
C
C THIS PART OF ROUTINE  COMPUTES  MDOTS
C
      ENTRY  SQAEROM
      DO 1000 N=1,L
C
C COMPUTE  MSDOT--- MASS LOSS RATE DUE TO SUBLIMATION
C
      IF (ASEXP ) 310,305,310
  305 MSDOT(N)=C.0
      GO TO 330
  310 BLOCK =-BSEXP/TTF(S,N)
      MSDOT(N)=   ASEXP *  PRELOC(N) **PSEXP * EXP(BLOCK)*R(1)**RIEXP
  330 COLL = (HE-HW(N))/(QCNET(N)*ELAM(N))
C
C COMPUTE  MCDOT--- MASS LOSS RATE DUE TO OXIDATION
C
C HALF ORDER OXIDATION
C
  380 IF (AEXP) 390,385,390
  385 MCDOT(N) =0.0
      GO TO 900
  390 MCDOT(N) = AEXP * EXP(-BEXP/TTF(S,N))
      IF (XORDER-0.5) 900,400,600
  400 ABC = 4.0* MCDCT(N)**2 * PRELOC(N) * CE * RSTO2
      PART = COLL * MCDOT(N)**2 * PRELOC(N) * RSTC2
      TEST = ABC/ PART**2
      IF (TEST.LT.7.E-12)GO TO  420
      MCDOT(N) =.5*((-PART) + SQRT (PART**2 + ABC))
      GO TO 900
  420 MCDOT(N) = CE /COLL
      GO TO 900
C
C FIRST ORDER OXIDATION
C
  600 MCDOT(N) = MCDOT(N)* PRELOC(N)* RSTO2 * CE/(1.0 + MCDOT(N)*PRELOC
     1 (N)* COLL*RSTO2)
C
C MDOT IS EQUAL TO THE LARGER OF MSDCT AND MCDOT
C
  900 IF (MCDOT(N).LT.MSDOT(N)) GO TO 950
      MDOT(N)= MCDOT(N)
      MSDCT(N)= 0.0
      GO TO 1000
  950 MDOT(N)= MSDOT(N)
      MCDOT(N)=C.0
 1000 CONTINUE
      RETURN
      END
```

Subroutine ADJUST.- Subroutine ADJUST computes the convective and radiant heating rates and the pressure and heating distributions to account for shape change. The flow chart for subroutine ADJUST is as follows:

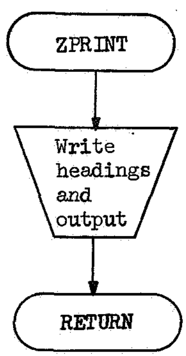


The program listing for subroutine ADJUST is as follows:

```
      SUBROUTINE ADJUST
C
C THIS ROUTINE ADJUSTS THE CONVECTIVE AND RADIANT HEATING RATES,THE PRESSURE
C AND HEATING DISTRIBUTION TO SHAPE CHANGE (ADJUST QC1,QR1,PRAT,QRAT )
C
      COMMON /PICK/ A(10,20),AA(20),AB(10,20),ALPHA(20),B(20),
     2 BS1(10,20),BS1B(10,20),C(10,20),CB(10,20),CK(10),CKETA(10,20),
     4 CKXI(10,20),COST(20),CP(10,20),D(10,20),DC(20),
     6 DELESQ,DELETA,DELTA(20),DELXI,DELXISQ,E(10,20),EIGHT3,
     8 ELAM(20),ETA(10),EXPG,F(10,20),GG,GIMACH,H1(10,20),H2(10,20),
     A H3(10,20),HC(20),HCOMB(20),HE,HW(20),IFIRST,IROCOL,ITC,ITR,ITT,
     C ITTO,LM1,LM2,MCDOT(20),MDOT(22),MSDOT(20),PID2,PRELOC(20),QC(20),
     E QC1,QCNET(20),QCOMB(20),QR(20),QR1,QS(20),RNS,RODPC,ROPCPP,
     G RSS(22),RSTO2,SIG,SIGDP,SIGMA,SIGP,SINT(20),SM1,SM2,TAU,TB,
     I TT(10,20),TTF(10,20),TWDELXI,TWOGI,V(20),VB(10),X(22),XDXISQ,
     K XCDXI,Y(10,20),Z(20),ZB(10)
      COMMON /INPUTS/ DUMMY(1),AEXP,ALCTAB(10),TTALC(10),MALPHC,NALPHC,
     2 ALPHAT(10),TALPHA(10),MALPHA,NALPHA,ALSTAB(10),TTALS(10),MALPHS,
     4 NALPHS,ASEXP,BETA,BEXP,BSEXP,CE,CKETATB(50),TTCKETA(10),
     6 ETATAB(5),NCKETA,NETA,CKXITAB(50),TTCKXI(10),XITAB(5),NCKXI,
     8 NXI,CORDSY,CPDP,CPP,CPTAB(10),TTABCP(10),MCP,NCP,DELTAO(20),
     A DELTAU,DELTMIN,DTMAX,ELAMTB(28),TTELAM(7),PELAM(4),NELAM,
     C NPELAM,ENDTAU,EPSONE,EPSONEP,EPSONPP,ERRORT,GAMBAR,GAMINF,
     E HCOMBTB(28),TTHCOMB(7),PHCOMB(4),NHCOMB,NPHCOMB,HCTAB(28),
     G TTABHC(7),PHC(4),NHC,NPHC,HETAB(10),TTABHE(10),MHE,NHE,HWTAB(15),
     I TTABHW(15),MHW,NHW,IADJUST,IPLOT,L,MACHNO,MAXITT,MDMAX,
     J MDOTO(20),
     K MWO2,MWSTR,NTP(7),PLTIME(15),PRAT(20),PRFREQ,PSEXP,PSTAGTB(10),
     M TTPSTAG(10),MPSTAG,NPSTAG,PTMAX,PTMIN,QCTAB(10),TTABQC(10),MQC,
     N NQC,QRAT(20),
     O QRRAT(20),QRTAB(10),TTABQR(10),MQR,NQR,R(20),RIEXP,RNSI,RO,RODP,
     Q ROP,RS(20),RSSMAX,S,STEBOL,T(10,20),TAUC,TBTAB(10),TTABTB(10),
     S MTB,NTB,TDPRIME,THETA(20),TPRIME,XO,XORDER,ZS(20),ZSMAX
      REAL MDOTC,MDOT,MCDOT,MSDOT,MWSTR,MWO2,MACHNO,MDMAX
      INTEGER S,SM1,SM2
      DIMENSION PSI(20)
      DIMENSION  UEUI(20), AL(20),AINT(20),YY(3)
      NSP1 = NSTEP + 1
      DO 50 N=1,L
      RSS(N) =   RS(N)  + DELTA(N)*COST(N)
   50 ZS(N) = ZS(N) + (DELTAO(N) - DELTA(N))* SINT(N)
      IF (IADJUST.EQ.0) RETURN
      RNS=(ZS(2)**2 + RSS(2)**2 -2.0*ZS(2)*ZS(1) + ZS(1)**2)/
     1(2.0*(ZS(2)-ZS(1)))
      SQRNS =  SQRT (RNS)
C ADJUST RATE TO SHAPE CHANGE
      QC1 = QC1 * SQRT ( RNSI/RNS )
      QR1 = QR1 * RNS/ RNSI
      PSI(1)=0.
      M=1
  100 DO 200 N=2,L
      NP1 = N+1
      NM1 = N-1
      IF (N.EQ. L) GO TO 130
      TANPHI = (RSS(NP1) -RSS(NM1))/(ZS(NP1)- ZS(NM1))
      GO TO 150
  130 TANPHI=  (RSS(L)-RSS(LM1))/(ZS(L)-ZS(LM1))
  150 PHI = ATAN (TANPHI)
      PSI(N)=PID2-PHI
  200 CONTINUE
C NEW PRESSURE  DISTRIBUTION
      DO 250 N=1,L
      PRAT(N) =(1.0 - GIMACH) *COS(PSI(N))**2   + GIMACH
      UEUI(N) = SQRT((1.0+ TWOGI) *(1.0-PRAT(N)**EXPG) )
  250 CONTINUE
C OBTAIN  NEW  HEAT DISTRIBUTION
C
```

```
C EVALUATE INTEGRAL AT L =0
      AL(1)=0.0
      AINTO=PRAT(1)*UEUI(1)* RSS(1)**2
  270 CONTINUE
      QRAT(1)=1.0
      DO 600 N=2,L
      NM1=N-1
      NM2 =N-2
      AINT=AINTO
      SUMH1=0.
      IF (N.EQ.2) GO TO 310
      DO 300 I=2,NM1
  300 SUMH1=SUMH1+H1(S,I)
  310 AL(N)= X(2) *(SUMH1 + (H1(S,1)+ H1(S,N))/2.0)
C
C EVALUATE INTEGRAL
C
      IF (N.EQ. 2)  GO TO 500
C EVALUATE Y(0),Y(1),Y(3)
      DO 400 K= 1,3
      NMK = N- (3-K)
  400 YY(K)= PRAT(NMK)*UEUI(NMK)*(RSS(NMK)**2)
      COEF2= AL(NM2)- AL(N)
      POXO= (AL(NM2)- AL(NM1))* COEF2
      P1X1=(AL(NM1)- AL(NM2))* (AL(NM1)- AL(N))
      P2X2= (AL(N)- AL(NM2)) *  (AL(N)-AL(NM1))
      COEF1= (3.0* AL(NM1)-2.0* AL(NM2) - AL(N))/POXO
      COEF3=(2.0*AL(N) + AL(NM2)- 3.0* AL(NM1))/ P2X2
      AINT(N) =((AL(N)- AL(NM2))**2/6.0)* ( YY(1)*COEF1 + YY(2)*COEF2/
     1 P1X1 + YY(3)* COEF3 )
      IF (N.GT.3)  AINT (N) = AINT (NM2) + AINT(N)
      GO TO 590
C N= 2
  500 YY(2)= (PRAT(1)+ PRAT(2))*(UEUI(1)+ UEUI(2))*((RSS(1)+ RSS(2))/2.0
     1  )**2  /4.0
      YY(3)=   PRAT(2)* UEUI(2) *(RSS(2)**2)
      AINT(N)=AL(2)*(4.0* YY(2) + YY(3))/6.0
  590 ANUM=PRAT(N)*UEUI(N)*RSS(N) *SQRNS
      QRAT(N) = ANUM / (SQRT(AINT(N))* GG)
  600 CONTINUE
      RETURN
      END
```

Subroutine ZPRINT.- Subroutine ZPRINT writes the output data. The flow chart for subroutine ZPRINT is as follows:

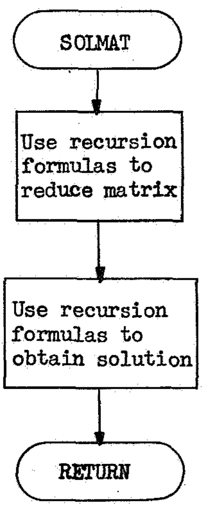


The program listing for subroutine ZPRINT is as follows:

```
      SUBROUTINE ZPRINT
C
      COMMON /PICK/ A(10,20),AA(20),AB(10,20),ALPHA(20),B(20),
     2 BS1(10,20),BS1B(10,20),C(10,20),CB(10,20),CK(10),CKETA(10,20),
     4 CKXI(10,20),COST(20),CP(10,20),D(10,20),DC(20),
     6 DELESQ,DELETA,DELTA(20),DELXI,DELXISQ,E(10,20),EIGHT3,
     8 ELAM(20),ETA(10),EXPG,F(10,20),GG,GIMACH,H1(10,20),H2(10,20),
     A H3(10,20),HC(20),HCOMB(20),HE,HW(20),IFIRST,IROCOL,ITC,ITR,ITT,
     C ITTO,LM1,LM2,MCDOT(20),MDOT(22),MSDOT(20),PID2,PRELOC(20),QC(20),
     E QC1,QCNET(20),QCOMB(20),QR(20),QR1,QS(20),RNS,RODPC,ROPCPP,
     G RSS(22),RSTO2,SIG,SIGDP,SIGMA,SIGP,SINT(20),SM1,SM2,TAU,TB,
     I TT(10,20),TTF(10,20),TWDELXI,TWOGI,V(20),VB(10),X(22),XDXISQ,
     K XODXI,Y(10,20),Z(20),ZB(10)
      COMMON /INPUTS/ DUMMY(1),AEXP,ALCTAB(10),TTALC(10),MALPHC,NALPHC,
     2 ALPHAT(10),TALPHA(10),MALPHA,NALPHA,ALSTAB(10),TTALS(10),MALPHS,
     4 NALPHS,ASEXP,BETA,BEXP,BSEXP,CE,CKETATB(50),TTCKETA(10),
     6 ETATAB(5),NCKETA,NETA,CKXITAB(50),TTCKXI(10),XITAB(5),NCKXI,
     8 NXI,CORDSY,CPDP,CPP,CPTAB(10),TTABCP(10),MCP,NCP,DELTAO(20),
     A DELTAU,DELTMIN,DTMAX,ELAMTB(28),TTELAM(7),PELAM(4),NELAM,
     C NPELAM,ENDTAU,EPSONE,EPSONEP,EPSONPP,ERRORT,GAMBAR,GAMINF,
     E HCOMBTB(28),TTHCOMB(7),PHCOMB(4),NHCOMB,NPHCOMB,HCTAB(28),
     G TTABHC(7),PHC(4),NHC,NPHC,HETAB(10),TTABHE(10),MHE,NHE,HWTAB(15),
     I TTABHW(15),MHW,NHW,IADJUST,IPLOT,L,MACHNO,MAXITT,MDMAX,
     J MDOTO(20),
     K MWO2,MWSTR,NTP(7),PLTIME(15),PRAT(20),PRFREQ,PSEXP,PSTAGTB(10),
     M TTPSTAG(10),MPSTAG,NPSTAG,PTMAX,PTMIN,QCTAB(10),TTABQC(10),MQC,
     N NQC,QRAT(20),
     O QRRAT(20),QRTAB(10),TTABQR(10),MQR,NQR,R(20),RIEXP,RNSI,RO,RODP,
     Q ROP,RS(20),RSSMAX,S,STEBOL,T(10,20),TAUO,TBTAB(10),TTABTB(10),
     S MTB,NTB,TOPRIME,THETA(20),TPRIME,XO,XORDER,ZS(20),ZSMAX
      REAL MDOTO,MDOT,MCDOT,MSDOT,MWSTR,MWO2,MACHNO,MDMAX
      INTEGER S,SM1,SM2
      DIMENSION QRR(20)
      EQUIVALENCE  (QRR(1),H1(1,1))
      DO 10 N=1,L
   10 QRR(N)= SIG * TTF(S,N)**4
      WRITE (6, 98)
   98 FORMAT ( *0*)
      WRITE (6,100) TAU,DELTAU
  100 FORMAT (*CTAU=*F10.4,14X*DELTAU=*F9.6)
      WRITE (6,101)  QC1, QR1, HE
  101 FORMAT (*C*14X*QC=*E11.4,5X,*QR=*E11.4,5X,*HE=*E11.4)
C
      WRITE (6,102) T(S,1)
  102 FORMAT (15X*T(S,1)=*E11.4)
      WRITE (6,105)
  105 FORMAT (*C*14X*TEMPERATURE (M,N)*)
      WRITE (6,110)(X (N),N=1,L)
  110 FORMAT (* ETA*6X*X=*15F8.5/(12X,15F8.5))
      DO 115 M=1,S
      MM= S- (M-1)
  115 WRITE  (6,120) ETA(MM),(TTF(MM,N),N=1,L)
  120 FORMAT (F6.3,6X15F8.1/(12X,15F8.1))
  140 FORMAT (* ETA*6X*X=*10(F9.5,3X)/(12X,10(F9.5,3X)))
  150 FORMAT (F6.3,6X10E12.4/(12X,10E12.4))
C
      WRITE (6,155)
  155 FORMAT (*C*14X*MDOT(N)--SURFACE MASS LOSS RATE*)
      WRITE (6,140) (X (N),N=1,L)
      WRITE (6,150) ETA(S),(MDOT(N),N=1,L)
C
      WRITE (6,165)
  165 FORMAT (*C*14X*MCDOT(N)--SURFACE MASS LOSS RATE DUE TO OXIDATION*)
      WRITE (6,150)  ETA(S),(MCDOT(N),N=1,L)
C
      WRITE (6,170)
  170 FORMAT (*C*14X*DELTA(N)--MATERIAL THICKNESS*)
      WRITE (6,150)  ETA(S), (DELTA(N),N=1,L)
```

```
C
      WRITE (6,175)
  175 FORMAT (*C*14X*QRAT(N)--RATIO OF LOCAL HEATING TC STAGNATION HEATI
     1NG*)
      WRITE (6,150) ETA(S),( QRAT(N),N=1,L)
C
      WRITE(6,176)
  176 FORMAT(*0*14X*PRAT(N)--RATIO OF LOCAL PRESS TO STAG PRESS*)
      WRITE(6,150) ETA(S),(PRAT(N),N=1,L)
C
      WRITE (6,180)
  180 FORMAT (*C*14X*QS(N)--NET HEAT INPUT*)
      WRITE (6,150) ETA(S), (QS    (N),N=1,L)
C
      WRITE (6,190)
  190 FCRMAT (*C*14X*QRR(N)--RERADIATION*)
      WRITE (6,150)  ETA(S), (QRR(N),N=1,L)
C
      WRITE (6,200)
  200 FORMAT (*C*14X*QCCMB(N)--HEAT DUE TO COMBUSTION FCR CXIDATION*)
      WRITE (6,150) ETA(S), (QCOMB(N),N=1,L)
C
      WRITE (6,400) ITC,ITR,ITTO ,IROCOL
  400 FCRMAT (*0      NO. ITER. COL.=*I4,5X,*NO. ITER. ROW=*I4,5X,*TOTAL
     1NO. ITER.=*I8,5X*IROCOL=*I3)
      RETURN
      END
```

Subroutine SOLMAT.- Subroutine SOLMAT solves a system of linear equations in which the matrix of coefficients is a tridiagonal matrix. The method of solution is equivalent to Gaussian elimination. The flow chart for subroutine SOLMAT is as follows:



The program listing for subroutine SOLMAT is as follows:

```
      SUBROUTINE SOLMAT(A,B,C,Z,V,D,T   ,N)
      DIMENSION W(20),SV(20),G(20),T(20),A(20),B(20),C(20),D(20)
      COMMON /HCLD/ TMIN
C
C THIS ROUTINE SOLVES THE TRIDIAGONAL (EXCEPT TWO ELEMENTS)  MATRIX
C
      W(1)=B(1)
      SV(1)= C(1) / B(1)
      X= Z/B(1)
      G(1)= D(1)/W(1)
      NM1=N-1
      NM2 = N-2
      DO 100 K=2,N
      KM1 = K-1
      IF (K.EQ.N) GO TO 20
      W(K) = B(K) - A(K)*SV(KM1)
      IF (K.EQ.2) GO TO 10
    4 SV(K)= C(K)/W(K)
    5 G(K) = (D(K)- A(K)*G(KM1))/W(K)
      GO TO 100
   10 SV(2) =(C(2)-X*A(2))/W(2)
      GO TO 5
   20 W(N)= B(N)- (A(N)- V*SV(NM2))*SV(NM1)
   30 G(N)=(D(N)-A(N)*G(KM1)-V*G(NM2)+V*SV(NM2)*G(KM1))/W(N)
  100 CONTINUE
      T(N)=G(N)
      DO 200 K=1,NM2
      KK= N-K
      T(KK)= G(KK)- SV(KK)*T(KK+1)
  200 CONTINUE
      T(1)= G(1)- SV(1)*T(2)- X*T(3)
      IF (TMIN.EQ.0.) RETURN
      DO 300 I=1,N
      IF(T(I) .LT.TMIN)  T(I)=TMIN
  300 CONTINUE
      RETURN
      END
```

## PROGRAM INPUT, OUTPUT, AND DIAGNOSTICS

### Input

Examples of input data are given in appendix B. The first card of the input is identification for the job. Any identification may be written in column 1 to and including column 80.

FORTRAN IV NAMELIST with the name D2430 is used to load the input data. Each input variable is initially set equal to zero by the program unless otherwise stated.

At least four inputs are associated with each table input: the dependent-table values, the independent-table values, the number of entries in the table, and the order of interpolation. The number of entries in the dependent and independent table must be the same. This is specified by a FORTRAN variable beginning with the letter N. The order of interpolation is a FORTRAN variable beginning with the letter M and may be 0, 1, or 2. For example, for first-order interpolation of the specific-heat array, set MCP=1; for second-order interpolation, set MCP=2. If the specific heat is a constant, set MCP=0.

The following list contains the input variables with the dimensions used in the program. The size of an array is limited to the dimensions stated. The maximum number of stations in the x-direction is 20 and the maximum number of stations in the y-direction is 10.

| FORTRAN variable | Symbol | Description |
|---|---|---|
| AEXP | $A_c$ | Coefficient of the exponential term when the Arrhenius expression is used for calculating MCDOT |
| ALCTAB(10) | $\alpha_c$ | Aerodynamic-blocking coefficient for heat and mass transfer associated with MCDOT, a function of time (TTALC) |
| ALPHAT(10) | $\alpha$ | Absorptance of surface, a function of temperature (TALPHA) |
| ALSTAB(10) | $\alpha_s$ | Aerodynamic-blocking coefficient for heat and mass transfer associated with MSDOT, a function of time (TTALS) |
| ASEXP | $A_s$ | Coefficient in the expression for calculating MSDOT |
| BETA | $\beta$ | Determines whether ablation or transpiration theory will be used for effect of mass transfer on heat transfer; for ablation theory, BETA=1; for transpiration theory, BETA=0 |
| BEXP | B | Power of the exponential term in the Arrhenius expression for calculating MCDOT |
| BSEXP | $B_s$ | Power of the exponential term in the expression for calculating MSDOT |
| CE | $C_e$ | Oxygen concentration, by mass, at edge of boundary layer |
| CKETATB(50) | $k_\eta$ | Thermal conductivity in $\eta$-direction, a function of $\eta$ (ETATAB) and temperature (TTCKETA) |
| CKXITAB(50) | $k_\xi$ | Thermal conductivity in $\xi$-direction, a function of $\xi$ (XITAB) and temperature (TTCKXI) |
| CORDSY | | Trigger to indicate coordinate system; if curvilinear coordinates, CORDSY=0; if Cartesian coordinates, CORDSY=1 |

| FORTRAN variable | Symbol | Description |
|---|---|---|
| CPDP | $c_p''$ | Specific heat of layer along y=0 |
| CPP | $c_p'$ | Specific heat of layer along x=L |
| CPTAB(10) | $c_p$ | Specific heat, a function of temperature (TTABCP) |
| DELTAO(20) | $\delta$ | Initial material thickness, must have L values |
| DELTAU | $\Delta\tau$ | Initial computing time interval |
| DELTMIN | | Minimum value allowed for DELTA |
| DTMAX | | Maximum DELTAU which can be used; if no value is given, DTMAX=2.0 |
| ELAMTB(28) | $\lambda$ | Ratio of mass of material removed per unit mass of oxygen that reaches the surface, a function of pressure (PELAM) and temperature (TTELAM) |
| ENDTAU | | Time at which calculation stops |
| EPSONE | $\epsilon$ | Emittance of front surface |
| EPSONEP | $\epsilon'$ | Emittance of layer along x=L |
| EPSONPP | $\epsilon''$ | Emittance of layer along y=0 |
| ERRORT | | Acceptable relative error in temperature |
| ETATAB(5) | $\eta$ | ETA table for CKETATB |
| GAMBAR | | Mean ratio of specific heats behind bow shock wave, used only in computation of heating-rate distribution around body |
| GAMINF | | Ratio of specific heats in free stream, used only in computing heating-rate distribution around body |
| HCOMBTB(28) | $\Delta H_c$ | Heat of combustion, a function of pressure (PHCOMB) and temperature (TTHCOMB) |
| HCTAB(28) | $\Delta H_s$ | Heat of sublimation, a function of pressure (PHC) and temperature (TTABHC) |
| HETAB(10) | $H_e$ | Total free-stream enthalpy, a function of time (TTABHE) |
| HWTAB(15) | $H_w$ | Enthalpy of gas at the wall temperature, a function of temperature (TTABHW) |

| FORTRAN variable | Symbol | Description |
|---|---|---|
| IADJUST | | Trigger for adjusting heating-rate and pressure distributions to shape change; if IADJUST=0, QRAT and PRAT are not adjusted; if IADJUST≠0, QRAT and PRAT will be adjusted to shape change |
| IPLOT | | Trigger for plotting routine; if IPLOT=0, no plots; if IPLOT≠0, the following plots will be made: RSS versus ZS at times indicated in PLTIME table; MDOT versus x at each PRFREQ time; and T(M,N) versus x indicated in NTP array at each PREREQ |
| L | | Number of stations in the x-direction |
| MACHNO | | Free-stream Mach number |
| MALPHA | | Order of interpolation for ALPHAT |
| MALPHC | | Order of interpolation for ALCTAB |
| MALPHS | | Order of interpolation for ALSTAB |
| MAXITT | | Maximum iteration count; when iteration count exceeds this number, DELTAU will be halved until DELTAU is less than 1.0E-6, then the program will stop and a message will be printed |
| MCP | | Order of interpolation for CPTAB |
| MDMAX | | Maximum expected MDOT; this must be given to get a reasonable scale for plots; not needed if IPLOT=0 |
| MDOTO(20) | $\dot{m}$ | Initial mass loss rate at surface, must have L values |
| MHE | | Order of interpolation for HETAB |
| MHW | | Order of interpolation for HWTAB |
| MPSTAG | | Order of interpolation for PSTAGTB |
| MQC | | Order of interpolation for QCTAB |
| MQR | | Order of interpolation for QRTAB |

| FORTRAN variable | Symbol | Description |
|---|---|---|
| MTB | | Order of interpolation for TBTAB |
| MWO2 | $M_{O_2}$ | Molecular weight of diatomic oxygen used in oxidation equation |
| MWSTR | $M_w$ | Molecular weight of free stream used in oxidation equation |
| NALPHA | | Number of entries in ALPHAT |
| NALPHC | | Number of entries in ALCTAB |
| NALPHS | | Number of entries in ALSTAB |
| NCKETA | | Number of entries in CKETATB |
| NCKXI | | Number of entries in CKXITAB |
| NCP | | Number of entries in CPTAB |
| NELAM | | Number of entries in ELAMTB |
| NETA | | Number of entries in ETATAB |
| NHC | | Number of entries in HCTAB |
| NHCOMB | | Number of entries in HCOMBTB |
| NHE | | Number of entries in HETAB |
| NHW | | Number of entries in HWTAB |
| NPELAM | | Number of entries in PELAM |
| NPHC | | Number of entries in PHC |
| NPHCOMB | | Number of entries in PHCOMB |
| NPSTAG | | Number of entries in PSTAGTB |
| NQC | | Number of entries in QCTAB |
| NQR | | Number of entries in QRTAB |
| NTB | | Number of entries in TBTAB |
| NTP(7) | | Array of seven values which specify the temperatures to be plotted; NTP(1) = the number of temperature rows to be plotted (may be six or less); NTP(2) through NTP(7), the row number of the temperatures to be plotted. For example, |

| FORTRAN variable | Symbol | Description |
|---|---|---|
| | | NTP(1)=3, NTP(2)=1, NTP(3)=5, NTP(4)=10, specifies that three (3) rows of temperature will be plotted and these rows are 1, 5, and 10 |
| NXI | | Number of entries in XITAB |
| PELAM(4) | | Pressure table for ELAMTB |
| PHC(4) | | Pressure table for HCTAB |
| PHCOMB(4) | | Pressure table for HCOMBTB |
| PLTIME(15) | | Times at which RSS versus ZS, that is, the body shape, will be plotted; not needed if IPLOT=0 |
| PRAT(20) | | Initial ratio of local to stagnation pressure, must have L values, not needed if IADJUST≠0 |
| PRFREQ | | Printing time frequency for output data |
| PSEXP | $p$ | Exponent of pressure term in sublimation equation |
| PSTAGTB(10) | | Stagnation pressure, a function of time (TTPSTAG) |
| PTMAX | | Maximum expected value of T, used to get reasonable scale in plotting, not needed if IPLOT=0 |
| PTMIN | | Minimum expected value of T, used to get reasonable scale in plotting, not needed if IPLOT=0 |
| QCTAB(10) | $q_C$ | Cold-wall convective heating rate, a function of time (TTABQC) |
| QRAT(20) | | Initial convective heating-rate distribution must have L values, not needed if IADJUST≠0 |
| QRRAT(20) | | Radiant heating-rate distribution over body, must have L values |
| QRTAB(10) | $q_r$ | Radiant heating-rate tables, a function of time (TTABQR) |
| R(20) | $R$ | Radius of curvature of base curve at node points, must have L values |
| RIEXP | $r$ | Exponent of nose-radius term in MSDOT equation |
| RNSI | $R_{stag}$ | Initial nose radius |

| FORTRAN variable | Symbol | Description |
|---|---|---|
| RO | $\rho$ | Material density |
| RODP | $\rho''$ | Density of layer along y=0 |
| ROP | $\rho'$ | Density of layer along x=L |
| RS(20) | $R_{cyl}$ | Cylindrical radius from body axis of symmetry to node points on the base curve, must have L values |
| RSSMAX | | Maximum expected value of RSS, used to get a reasonable scale for plots, not needed if IPLOT=0 |
| S | | Number of stations in y-direction |
| STEBOL | $\sigma$ | Stefan-Boltzmann constant for radiation |
| T(10,20) | | Initial temperature, must have S*L values |
| TALPHA(10) | | Temperature table for ALPHAT |
| TAUO | $\tau$ | Initial time |
| TBTAB(10) | | Temperature to which back surface is radiating, a function of time (TTABTB) |
| TDPRIME | | Thickness of layer along y=0 |
| THETA(20) | $\theta$ | Angle (in degrees) less than or equal to $90^{o}$ between RS and R, must have L values |
| TMIN | | Minimum temperature value; if TMIN≠0 and a computed temperature goes below TMIN, the temperature will be set equal to TMIN; if TMIN=0, no restraint will be made on the computed temperatures |
| TPRIME | | Thickness of layer along x=L |
| TTABCP(10) | | Temperature table for CPTAB |
| TTABHC(10) | | Temperature table for HCTAB |
| TTABHE(10) | | Time table for HETAB |
| TTABHW(15) | | Temperature table for HWTAB |
| TTABQC(10) | | Time table for QCTAB |

| FORTRAN variable | Symbol | Description |
|---|---|---|
| TTABQR(10) | | Time table for QRTAB |
| TTABTB(10) | | Time table for TBTAB |
| TTALC(10) | | Time table for ALCTAB |
| TTALS(10) | | Time table for ALSTAB |
| TTCKETA(10) | | Temperature table for CKETATB |
| TTCKXI(10) | | Temperature table for CKXITAB |
| TTELAM(7) | | Temperature table for ELAMTB |
| TTHCOMB(7) | | Temperature table for HCOMBTB |
| TTPSTAG(10) | | Time table for PSTAGTB |
| XITAB(5) | $\xi$ | Table of values of CKXITAB |
| XO | $x_b$ | Length of base curve |
| XORDER | | Order of oxidation |
| ZS(20) | | Initial distance from the initial stagnation point to RSS along body axis of symmetry, must have L values |
| ZSMAX | | Maximum expected value of ZS, used to get reasonable scale for plotting RSS versus ZS, not needed if IPLOT=0 |

## Output

Examples of output data are given in appendix B. The input data are printed at the beginning of the output listing in the same order in which they appear in the NAMELIST statement. Then the identification card is printed. Headings and interpretations are as follows:

| Heading | Description |
|---|---|
| TAU | Time at which the calculations were made |
| DELTAU | The computing time interval |
| QC | Convective heating rate |
| QR | Radiant heating rate |

Heading

| | |
|---|---|
| HE | Total free-stream enthalpy |
| T(S,1) | Temperature at time $\tau - \Delta\tau$; this value can indicate whether a reasonable $\Delta\tau$ is being used; by observing this value and the value at $\tau$, unusual behavior might indicate the need for a smaller $\Delta\tau$. |
| TEMPERATURE (M,N) | Temperatures; to locate the station read ETA to the left and x above the temperature column; up to 15 temperatures are printed on one line; if more columns have been used, the remaining temperatures will be printed on the next line |
| ETA | Dimensionless y values, printed in the first column on the left side of the page |
| X | Length along base curve from stagnation point to the station, printed in the second column and reading from left to right |
| MDOT(N) | Surface mass loss rate at station n |
| MCDOT(N) | Mass loss due to oxidation at station n |
| DELTA(N) | Material thickness at station n |
| QRAT(N) | Ratio of local heating to stagnation heating at station n |
| PRAT(N) | Ratio of local pressure to stagnation pressure at station n |
| QS(N) | Net heat input at station n |
| QRR(N) | Reradiation at station n |
| QCOMB(N) | Heat due to combustion for oxidation at station n |
| NO.ITER.COL. | Number of iterations for the previous column solution |
| NO.ITER.ROW. | Number of iterations for the previous row solution |
| TOTAL NO.ITER. | Total number of iterations from the beginning of the problem |
| IROCOL | Tells at which solution the printout was made; value of 1 indicates column solution; 2, row solution |

## Diagnostics

The program has several automatic stops to avoid the waste of computer time on problems which appear to be having computational difficulties. These stops are

(1) DELTA < DELTMIN: If any thickness DELTA becomes less than the input DELTMIN a normal printout is made and the program will stop.

(2) Negative temperature: If any temperature becomes negative, a normal printout is made and the program will stop.

(3) DELTAU < 1.0E-6: If the computing time interval DELTAU becomes less than 1.0E-6, the message TEMPERATURE ITERATION DOES NOT CONVERGE will be printed. The current estimated temperatures are printed, a normal printout is made, and the program will stop.

(4) Iteration count exceeded: If the maximum iteration count input MAXITT is exceeded and the calculation is a row solution, the computing interval cannot be halved. The message THIS IS A ROW SOLUTION, DELTAU CANNOT CHANGE is printed. The current estimated temperatures are printed, a normal printout is made, and the program will stop.

(5) Temperature diverging: If any temperature begins diverging, the message TEMPERATURE IS DIVERGING . . . . . WHY is printed. The current estimated temperatures are printed, a normal printout is made, and the program will stop.

Whenever these diagnostics appear, the input should be checked to make sure that all initial conditions have been given. Check all input tables for any discontinuities. Negative temperatures may result from oscillations caused by time intervals which are too large. High values of MDOT and rapid changes of heat input with time may require smaller time intervals for computational purposes.

## SAMPLE CASES

Three sample cases are presented to illustrate the operation of the computer program. All the cases are for ablating bodies of different geometries: a hemisphere, a hemispherically blunted cone, and a right-circular cylinder. A listing of the input data and a sample of the output data for each case are shown in appendix B.

Computer-generated curves of some of the output from the sample cases are shown in figures 1, 2, and 3. The curves show body shape change due to ablation, histories of mass-transfer rate over the surface of the bodies, and selected temperature histories. The body shape is plotted at each time listed in the input PLTIME. The mass-transfer rates over the surface and the temperatures along the rows specified by the input NTP are plotted at each printing frequency for the output data.

The computing time depends on the accuracy desired; the boundary condition, that is, the heating-rate history; and the number of node points. The computational times for the sample cases are 136 seconds for the hemisphere, 312 seconds for the right-circular-cylinder, and 150 seconds for the hemispherically blunted cone. These cases have not been optimized with respect to time and, therefore, may run in shorter periods of time.

Langley Research Center,
National Aeronautics and Space Administration,
Hampton, Va., September 3, 1971.

# APPENDIX A

## LANGLEY LIBRARY SUBROUTINES

### Subroutine FTLUP

Language: FORTRAN

Purpose: Computes $y = F(x)$ from a table of values using first- or second-order interpolation. An option to give $y$ a constant value for any $x$ is also provided.

Use: CALL FTLUP(X, Y, M, N, VARI, VARD)

X The name of the independent variable $x$.

Y The name of the dependent variable $y = F(x)$.

M The order of interpolation (an integer)

$M = 0$ for $y$ a constant. VARD(I) corresponds to VARI(I) for $I = 1, 2, \ldots, N$. For $M = 0$ or $N \leqq 1$, $y = F(\text{VARI}(1))$ for any value of $x$. The program extrapolates.

$M = 1$ or 2. First or second order if VARI is strictly increasing (not equal).

$M = -1$ or $-2$. First or second order if VARI is strictly decreasing (not equal).

N The number of points in the table (an integer).

VARI The name of a one-dimensional array which contains the N values of the independent variable.

VARD The name of a one-dimensional array which contains the N values of the dependent variable.

Restrictions: All the numbers must be floating point. The values of the independent variable $x$ in the table must be strictly increasing or strictly decreasing. The following arrays must be dimensioned by the calling program as indicated: VARI(N), VARD(N).

Accuracy: A function of the order of interpolation used.

References: (a) Nielsen, Kaj L.: Methods in Numerical Analysis. The Macmillan Co., c.1956, pp. 87-91.
(b) Milne, William Edmund: Numerical Calculus. Princeton Univ. Press, c.1949, pp. 69-73.

Storage: $430_8$ locations.

Error condition: If the VARI values are not in order, the subroutine will print TABLE BELOW OUT OF ORDER FOR FTLUP AT POSITION xxx TABLE IS STORED IN LOCATION xxxxxx (absolute). It then prints the contents of VARI and VARD, and STOPS the program.

Subroutine date: September 12, 1969.

## Subroutine DISCOT

Language: FORTRAN

Purpose: DISCOT performs single or double interpolation for continuous or discontinuous functions. Given a table of some function y with two independent variables, x and z, this subroutine performs $K_x$th- and $K_z$th-order interpolation to calculate the dependent variable. In this subroutine all single-line functions are read in as two separate arrays and all multiline functions are read in as three separate arrays; that is,

$$x_i \qquad (i = 1, 2, \ldots, L)$$

$$y_j \qquad (j = 1, 2, \ldots, M)$$

$$z_k \qquad (k = 1, 2, \ldots, N)$$

Use: CALL DISCOT (XA, ZA, TABX, TABY, TABZ, NC, NY, NZ, ANS)

XA The x argument

ZA The z argument (may be the same name as x on single lines)

TABX A one-dimensional array of x values

TABY A one-dimensional array of y values

TABZ A one-dimensional array of z values

NC A control word that consists of a sign (+ or -) and three digits. The control word is formed as follows:

(1) If NX = NY, the sign is negative. If NX ≠ NY, then NX is computed by DISCOT as NX = NY/Nz, and the sign is positive and may be omitted if desired.

(2) A one in the hundreds position of the word indicates that no extrapolation occurs above $z_{max}$. With a zero in this position, extrapolation occurs when $z > z_{max}$. The zero may be omitted if desired.

(3) A digit (1 to 7) in the tens position of the word indicates the order of interpolation in the x-direction.

(4) A digit (1 to 7) in the units position of the word indicates the order of interpolation in the z-direction.

NY The number of points in y array

NZ The number of points in z array

ANS The dependent variable y

The following programs will illustrate various ways to use DISCOT:

CASE I:

Given $y = f(x)$
NY = 50
NX (number of points in $x$ array) = NY
Extrapolation when $z > z_{max}$
Second-order interpolation in x-direction
No interpolation in z-direction
Control word = -020

```
  DIMENSION TABX (50), TABY (50)
1 FORMAT (8E 9.5)
  READ (5,1) TABX, TABY
  READ (5,1) XA
  CALL DISCOT (XA, XA, TABX, TABY, TABY, -020, 50, 0, ANS)
```

CASE II:

Given $y = f(x,z)$
NY = 800
NZ = 10
NX = NY/NZ (computed by DISCOT)
Extrapolation when $z > z_{max}$
Linear interpolation in x-direction
Linear interpolation in z-direction
Control word = 11

```
  DIMENSION TABX (800), TABY (800), TABZ (10)
1 FORMAT (8E 9.5)
  READ (5,1) TABX, TABY, TABZ
  READ (5,1) XA, ZA
  CALL DISCOT (XA, ZA, TABX, TABY, TABZ, 11, 800, 10, ANS)
```

CASE III:

Given $y = f(x,z)$
NY = 800
NZ = 10
NX = NY
Extrapolation when $z > z_{max}$
Seventh-order interpolation in x-direction
Third-order interpolation in z-direction
Control word = -73

```
  DIMENSION TABX (800), TABY (800), TABZ (10)
1 FORMAT (8E 9.5)
  READ (5,1) TABX, TABY, TABZ
  READ (5,1) XA, ZA
  CALL DISCOT (XA, ZA, TABX, TABY, TABZ, -73, 800, 10, ANS)
```

CASE IV:

Same as Case III with no extrapolation above $z_{max}$. Control word = -173

```
  CALL DISCOT (XA, ZA, TABX, TABY, TABZ, -173, 800, 10, ANS)
```

Restrictions: See rule (5c) of section "Method" for restrictions on tabulating arrays and discontinuous functions. The order of interpolation in the x- and z-directions may be from 1 to 7. The following subprograms are used by DISCOT: UNS, DISSER, LAGRAN.

Method: Lagrange's interpolation formula is used in both the x- and z-directions for interpolation. This method is explained in detail in reference (a) of this subroutine. For a search in either the x- or z-direction, the following rules are observed:

(1) If $x < x_1$, the routine chooses the following points for extrapolation:

$$x_1, x_2, \ldots, x_{k+1} \quad \text{and} \quad y_1, y_2, \ldots, y_{k+1}$$

(2) If $x > x_n$, the routine chooses the following points for extrapolation:

$$x_{n-k}, x_{n-k+1}, \ldots, x_n \quad \text{and} \quad y_{n-k}, y_{n-k+1}, \ldots, y_n$$

(3) If $x \leqq x_n$, the routine chooses the following points for interpolation:

When $k$ is odd,

$$x_{i-\frac{k+1}{2}}, x_{i-\frac{k+1}{2}+1}, \ldots, x_{i-\frac{k+1}{2}+k} \quad \text{and} \quad y_{i-\frac{k+1}{2}}, y_{i-\frac{k+1}{2}+1}, \ldots, y_{i-\frac{k+1}{2}+k}$$

When $k$ is even,

$$x_{i-\frac{k}{2}}, x_{i-\frac{k}{2}+1}, \ldots, x_{i-\frac{k}{2}+k} \quad \text{and} \quad y_{i-\frac{k}{2}}, y_{i-\frac{k}{2}+1}, \ldots, y_{i-\frac{k}{2}+k}$$

(4) If any of the subscripts in rule (3) become negative or greater than $n$ (number of points), rules (1) and (2) apply. When discontinuous functions are tabulated, the independent variable at the point of discontinuity is repeated.

(5) The subroutine will automatically examine the points selected before interpolation and if there is a discontinuity, the following rules apply. Let $x_d$ and $x_{d+1}$ be the point of discontinuity.

(a) If $x \leqq x_d$, points previously chosen are modified for interpolation as shown:

$$x_{d-k}, x_{d-k+1}, \ldots, x_d \quad \text{and} \quad y_{d-k}, y_{d-k+1}, \ldots, y_d$$

(b) If $x > x_d$, points previously chosen are modified for interpolation as shown:

$$x_{d+1}, x_{d+2}, \ldots, x_{d+k} \quad \text{and} \quad y_{d+1}, y_{d+2}, \ldots, y_{d+k}$$

(c) When tabulating discontinuous functions, there must always be $k + 1$ points above and below the discontinuity in order to get proper interpolation.

(6) When tabulating arrays for this subroutine, both independent variables must be in ascending order.

(7) In some engineering programs with many tables, it is quite desirable to read in one array of x values that could be used for all lines of a multiline function or different functions. Even though this situation is not always applicable, the subroutine has been written to handle it. This procedure not only saves much time in preparing tabular data, but also can save many locations previously used when every y coordinate had to have a corresponding x coordinate. Another additional feature that may be useful is the possibility of a multiline function with no extrapolation above the top line.

Accuracy: A function of the order of interpolation used.

Reference: (a) Nielsen, Kaj L.: Methods in Numerical Analysis. The Macmillan Co., c.1956.

Storage: $555_8$ locations.

Subprograms used:

| | | |
|---|---|---|
| UNS | $40_8$ | locations. |
| DISSER | $110_8$ | locations. |
| LAGRAN | $55_8$ | locations. |

Subroutine date: August 1, 1968.

# APPENDIX B

## SAMPLE LISTINGS

This appendix gives sample input and output listings for three sample cases. The sample input listing for a teflon hemisphere is given below:

```
1 TEFLON HEMISPHERE
$D2430
ALCTAB=1.,       ALPHAT=1.,       ALSTAB=.4,
CE=.232,
GAMBAR=1.4,   GAMINF=1.4,   MACHNO=2.6,
NHE=4, MHE=1, TTABHE=0,18,18.1,20, HETAB=2*.4881E+7,2*.2906E+6,
NHW=14, MHW=1, TTABHW=277.8,555.6,833.3,1111.1,1388.9,1666.7,1944.4,2222.2,
2500,2777.8,3055.6,3333.3,3611.1,3888.9,
HWTAB=-23240.,.258E+6,.5555E+6,.8717E+6,.12071E+7,.15457E+7,.1901E+7,.22779E+7,
.27265E+7,.3217E+7,.3951E+7,.4788E+7,.5927E+7,.6973E+7,
MWO2=32,  MWSTR=32,
PSTAGTB=.95,
NQC=5,  MQC=1, TTABQC=0,1,18,18.5,20,  QCTAB=410,2*.16E+7,2*410,
ASFXP=.11732E+9,    BSFXP=20400.,
NETA=2,  ETATAB=0.,.5.,   TTCKETA=316.7,600.,    NCKETA=4,
CKETATB=.2768,.38098,.27685,.38098,
NXI=2,  XITAB=0.,.5., TTCKXI=316.7,600., NCKXI=4,
CKXITAB=.27685,.38098,.27685,.38098,
NCP=2, MCP=1, TTABCP=277.8,388.9,  CPTAB=1004.,1087.8,
NPELAM=4, PELAM=.01,.1,1.,10., TTELAM=55.6,111.,166.7,222.,277.8,333.3,388.9,
NELAM=28, ELAMTB=28*.75,
NPHCOMB=4, PHCOMB=.01,.1,1.,10., TTHCOMB=55.6,111.,166.7,222,277.8,333.3,388.9,
NHCOMB=28, HCOMBTB=28*0,
NPHC=4, PHC=.01,.1,1.,10., TTABHC=277.8,388.9,500,555.6,611,666.7,722,
NHC=28, HCTAB=28*.16271E+7,
RO=2163,
T=100*300,
DELTAU=.48828125E-3,  DTMAX=.015625,     ENDTAU=8.,  PRFREQ=1,
ERRORT =.0001,    MAXITT=5,
IADJUST=1,
IPLOT=1,  MDMAX=.4, PTMAX=1200,  PTMIN=200,  RSSMAX=.01,   ZSMAX=.01,
NTP=2,5,10,  PLTIME=0,4.,8.,
DELTAO=10*.007315,  DELTMIN=.3048E-4,
L=10,   S=10,   XO=.00047884,
R=10*.3048E-3, RS=0,.5291E-4,.1042E-3,.1524E-3,.1959E-3,.2335E-3,.264E-3,
.2864E-3,.3002E-3,.3048E-3,   THETA=90,80,70,60,50,40,30,20,10,0,
ZS=0,.11576E-3,.45953E-3,.0010219,.0017827,.0027219,.0038099,.0050137,
.0062966,.0076198,
RNSI=.00762,
TBTAB=300,    $
```

The sample output listing for a teflon hemisphere is given below:

```
TEFLON HEMISPHERE


TAU=     1.0142              DELTAU=  .015625

                 QC= 1.5828E+06     QR= 0.              HE= 4.8810E+06
                 T(S,1)= 1.0109E+03

                 TEMPERATURE (M,N)
ETA        X=  0.00000  .00005  .00011  .00016  .00021  .00027  .00032  .00037  .00043  .00048
1.000          1016.8  1000.2   978.4   931.2   860.7   770.5   668.5   566.6   485.4   456.6
 .889           369.5   368.1   364.9   358.7   350.3   340.4   330.1   320.7   313.8   311.5
 .778           303.7   303.6   303.5   303.1   302.7   302.1   301.6   301.1   300.7   300.6
 .667           300.2   300.2   300.1   300.1   300.1   300.1   300.1   300.0   300.0   300.0
 .556           300.0   300.0   300.0   300.0   300.0   300.0   300.0   300.0   300.0   300.0
 .444           300.0   300.0   300.0   300.0   300.0   300.0   300.0   300.0   300.0   300.0
 .333           300.0   300.0   300.0   300.0   300.0   300.0   300.0   300.0   300.0   300.0
 .222           300.0   300.0   300.0   300.0   300.0   300.0   300.0   300.0   300.0   300.0
 .111           300.0   300.0   300.0   300.0   300.0   300.0   300.0   300.0   300.0   300.0
0.000           300.0   300.0   300.0   300.0   300.0   300.0   300.0   300.0   300.0   300.0

                 MDOT(N)--SURFACE MASS LOSS RATE
ETA        X=  0.00000     .00005      .00011      .00016      .00021      .00027      .00032      .00037      .00043      .00048

1.000          2.2692E-01  1.6261E-01  1.0336E-01  3.5873E-02  5.9613E-03  3.7163E-04  6.5477E-06  2.7122E-08  6.5565E-11  4.6192E-12


                 MCDOT(N)--SURFACE MASS LOSS RATE DUE TO OXIDATION
1.000          0.          0.          0.          0.          0.          0.          0.          0.          0.          0.


                 DELTA(N)--MATERIAL THICKNESS
1.000          7.3075E-03  7.3105E-03  7.3124E-03  7.3142E-03  7.3149E-03  7.3150E-03  7.3150E-03  7.3150E-03  7.3150E-03  7.3150E-03


                 QRAT(N)--RATIO OF LOCAL HEATING TO STAGNATION HEATING
1.000          1.0000E+00  9.9094E-01  9.3792E-01  8.3942E-01  7.1607E-01  5.7410E-01  4.2728E-01  2.9331E-01  1.9487E-01  1.6190E-01


                 PRAT(N)--RATIO OF LOCAL PRESS TO STAG PRESS
1.000          1.0000E+00  9.7350E-01  8.9578E-01  7.7694E-01  6.3093E-01  4.7522E-01  3.2924E-01  2.1028E-01  1.3263E-01  1.1245E-01


                 QS(N)--NET HEAT INPUT
1.000          8.1363E+05  9.7746E+05  1.0515E+06  1.0768E+06  9.8590E+05  8.1712E+05  6.2379E+05  4.3858E+05  2.9663E+05  2.4798E+05


                 QRR(N)--RERADIATION
1.000          0.          0.          0.          0.          0.          0.          0.          0.          0.          0.


                 QCCMB(N)--HEAT DUE TO COMBUSTION FOR OXIDATION
1.000          0.          0.          0.          0.          0.          0.          0.          0.          0.          0.

     NO. ITER. COL.=   2     NO. ITER. ROW=   1     TOTAL NO. ITER.=     133     IROCOL=  1
```

The sample input listing for a graphite hemisphere-cone is given below:

```
1 GRAPHITE HEMISPHERE - 30 DEG. CONE
 $D2430
  ALCTAB=1.,    ALPHAT=1.,        ALSTAB=1.,
  CF=.232,
  GAMBAR=1.4,    GAMINF=1.4,
  MACHNO=20.,
  HETAB=.92976E+8,    TTABHE=100.,
  NHW=14, MHW=1, HWTAB=-.2324E+5,.258E+5,.5555E+6,.8717E+6,.12071E+7,.15457E+7,
  .1901E+7,.22779E+7,.27265E+7,.3217E+7,.3951E+7,.4788E+7,.5927E+7,.6973E+7,
  TTABHW=277.8,555.6,833.3,1111.1,1388.9,1666.7,1944.4,2222.2,2500.,2777.8,
  3055.6,3333.3,3611.1,3888.9,
  MWO2=32.,    MWSTR=32.,
  PSTAGTB=1.,
  NQC=3,  MQC=1, QCTAB=.408E+4,.28E+8,.28006E+8,   TTABQC=0,2,1000.,
  EPSONE=.98,     STEBOL=.56697E-7,
  QRRAT=1,.5,
  NQR=3,    MQR=1,  TTABQR=0,3,1000,    QRTAB=0,2*.1135E+8,
  AFXP=.48825E+11,        BFXP=.425E+5,     ASFXP=273560.,    BSFXP=61670.,
  XORDER=1.,
  NCKETA=20,    CKETATB=168.4,103.5,85.4,74.8,62.35,54.87,52.38,51.1,50.51,
  49.26,168.4,103.5,85.4,74.8,62.35,54.87,52.38,51.1,50.51,49.26,
  NETA=2,    ETATAB=0,5,    TTCKETA=277.8,555.6,694.4,833.3,1111.1,1388.9,
  1666.7,1944.4,2222.2,3333.3,
  NCKXI=20,    CKXITAB=168.4,103.5,85.4,74.8,62.35,54.87,52.38,51.1,50.51,49.26,
  168.4,103.5,85.4,74.8,62.35,54.87,52.38,51.1,50.51,49.26,
  NXI=2,    XITAB=0,5.,    TTCKXI=277.8,555.6,694.4,833.3,1111.1,1388.9,1666.7,
  1944.4,2222.2,3333.3,
  NCP=9,   MCP=1, CPTAB=669.4,1046.,1297.,1506.,1673.6,1841.,1966.,2092.,2218.,
  TTABCP= 277.8,416.7,555.6,694.4,833.3,1111.1,1388.9,1944.4,2777.8,
  NFLAM=8,    ELAMTB=28*.75,    NPELAM=4,   PELAM=.01,.1,1.,10.,
  TTFLAM=277.8,555.6,833.3,1111.1,1388.9,1666.7,1944.4,
  NHCOMB=28,    HCOMBTB=.9553E+7,.99158E+7,.10353E+8,.11322E+8,.14562E+8,
  .23755E+8,.31472E+8,.9553E+7,.99158E+7,.10353E+8,.11322E+8,.14562E+8,
  .23755E+8,.31472E+8,.9553E+7,.99158E+7,.10353E+8,.11322E+8,.14562E+8,
  .23755E+8,.31472E+8,.9553E+7,.99158E+7,.10353E+8,.11322E+8,.14562E+8,
  .23755E+8,.31472E+8,    NPHCOMB=4, PHCOMB=.1,1.,10.,100.,
  TTHCOMB=1000.,1500.,2000.,2500.,3000.,3500.,4000.,
  NHC=28, HCTAB=28*.27893E+8, NPHC=4, PHC=.01,.1,1.,10.,
  TTABHC=277.8,555.6,833.3,1111.1,2222.2,3333.3,3888.9,
  RO=1698.,
  T=200*300,
  DELTAO= 10*.01905,     DELTMIN=.1E-7,      XO=.4645,
  L=10,    S=5,
  R=4*.5,6*.1E+49,
  RS=0,.05063,.09488,.12951,.15542,.18123,.20705,.23284,.25865,.28447,
  THETA=90,70.6,51.2,31.8,6*30,
  PSFXP=-.17,      RIFXP=.5,     RNSI=.17145,
  ZS=0,.009735,.037271,.0811,.12629,.171,.21568,.2603,.3048,.34973,
  DELTAU=.0078125,     DTMAX=.0625,     ENDTAU=60.,   PRFREQ=4.,
  ERRORT=.001,    MAXITT=5,
  IADJUST=1,
  IPLOT=1, MDMAX=.4, NTP=2,1,5,  PLTIME=0,15,30,45,60, PTMAX=5000, RSSMAX=.4,
  ZSMAX=.4,
 $
```

The sample output listing for a graphite hemisphere-cone is given below:

```
GRAPHITE HEMISPHERE-30 DEG. CONE


TAU=     4.1016              DELTAU=  .062500

             QC= 2.7956E+C7     QR= 1.1354E+07      HE= 9.2976E+07
             T(S,1)= 4.0961E+03

             TEMPERATURE (M,N)
ETA        X= 0.00000  .05161  .10322  .15483  .20644  .25806  .30967  .36128  .41289  .46450
1.000          4087.5  3800.9  2913.6  2084.9  1649.4  1554.2  1478.2  1415.5  1356.4  1306.0
 .750          2432.6  2263.4  1768.4  1332.6  1101.4  1049.2  1009.3   977.0   947.2   922.0
 .500          1473.2  1397.6  1162.0   949.7   826.1   797.4   775.7   758.0   741.4   727.4
 .250          1058.2  1022.4   898.6   778.1   700.4   681.6   667.6   656.0   645.1   636.0
0.000           944.7   918.0   823.6   727.6   663.4   647.4   635.6   625.7   616.4   608.7

             MDOT(N)--SURFACE MASS LOSS RATE
ETA        X=  0.00000      .05161      .10322      .15483      .20644      .25806      .30967      .36128      .41289      .46450

1.000          5.4218E-02  4.6C94E-02  3.5134E-02  2.1437E-02  8.6107E-03  3.0432E-03  8.7433E-04  2.5633E-04  6.9847E-05  2.0802E-05


             MCDOT(N)--SURFACE MASS LCSS RATE DUE TO OXIDATION
1.000          0.          4.6C94E-02  3.5134E-02  2.1437E-02  8.6107E-03  3.0432E-03  8.7433E-04  2.5633E-04  6.9847E-05  2.0802E-05


             DELTA(N)--MATERIAL THICKNESS
1.000          1.8974E-02  1.8979E-02  1.9003E-02  1.9032E-02  1.9048E-02  1.9050E-02  1.9050E-02  1.9050E-02  1.9050E-02  1.9050E-02


             QRAT(N)--RATIO CF LOCAL HEATING TO STAGNATION HEATING
1.000          1.0000E+00  9.6960E-01  7.3907E-01  4.5250E-01  3.2368E-01  2.9931E-01  2.8055E-01  2.6552E-01  2.5153E-01  2.3926E-01


             PRAT(N)--RATIO OF LOCAL PRESS TO STAG PRESS
1.000          1.0000E+00  8.9181E-01  6.0840E-01  3.4999E-01  2.5237E-01  2.5150E-01  2.5172E-01  2.5241E-01  2.5132E-01  2.4960E-01


             QS(N)--NET HEAT INPUT
1.000          3.2676E+07  2.9857E+07  1.8544E+07  1.1466E+07  8.5680E+06  8.1400E+06  7.7121E+06  7.3254E+06  6.9510E+06  6.6178E+06


             QRR(N)--RERADIATION
1.000          1.5510E+07  1.1597E+07  4.0042E+06  1.0498E+06  4.1125E+05  3.2420E+05  2.6528E+05  2.2308E+05  1.8807E+05  1.6166E+05


             QCOMB(N)--HEAT DUE TO COMBUSTION FOR OXIDATION
1.000          0.          1.3C99E+06  4.9219E+05  2.2534E+05  7.8413E+04  2.5029E+04  6.8012E+03  1.9368E+03  5.1694E+02  1.5195E+02

     NO. ITER. COL.=    2     NC. ITER. ROW=    1     TOTAL NO. ITER.=      134     IROCOL=  1
```

The sample input listing for a right-circular cylinder is given below:

```
1 SAMPLE PROBLEM IN CARTESIAN COORDINATES
 $D2430
 AEXP=.48825E+11,    BEXP=.425E+5,    ASEXP=.27356E+6,   BSEXP=61670.,  XORDER=1,
 NETA=2,    ETATAB=0,5.,   NCKETA=20,
 CKETATB=168.4,103.5,85.4,74.8,62.35,54.87,52.38,51.1,50.51,49.26,168.4,
 103.5,85.4,74.8,62.35,54.87,52.38,51.1,50.51,49.26,
 TTCKETA=277.8,555.6,694.4,833.3,1111.1,1388.9,1666.7,1944.4,2222.2,3333.3,
 NXI=2,   XITAB=0,5,   TTCKXI=277.8,555.6,694.4,833.3,1111.1,1388.9,1666.7,
 1944.4,2222.2,3333.3,    NCKXI=20,
 CKXITAB= 168.4,103.5,85.4,74.8,62.35,54.87,52.38,51.1,50.51,49.26,168.4,103.5,
 85.4,74.8,62.35,54.87,52.38,51.1,50.51,49.26,
 NCP=9, MCP=1,   TTABCP=277.8,416.7,555.6,694.4,833.3,1111.1,1388.9,1944.4,
 2777.8,
 CPTAB= 669.4,1046.,1297,1506,1673.6,1841,1966,2092,2218
 NPELAM=4, PELAM=.01,.1,1.,10,   NELAM=28, ELAMTB=28*.75,
  TTFLAM=277.8,555.6,833.3,1111.1,1388.9,1666.7,1944.4,
 NPHCOMB= 4, PHCOMB=1,1,10,100, TTHCOMB=1000,1500,2000,2500,3000,3500,4000,
 NHCOMB=28,   HCOMBTB=.9553E+7,.99158E+7,.10353E+8,.11322E+8,.14562E+8,.23755E+8,
 .31472E+8,.9553E+7,.99158E+7,.10353E+8,.11322E+8,.14562E+8,.23755E+8,.31472E+8,
 .9553E+7,.99158E+7,.10353E+8,.11322E+8,.14562E+8,.23755E+8,.31472E+8,.9553E+7,
 .99158E+7,.10353E+8,.11322E+8,.14562E+8,.23755E+8,.31472E+8,
 NPHC=4,   PHC=.01,.1,1,10,   NHC=28,   HCTAB=28*.27893E+8,   TTABHC=277.8,555.6,
 833.3,1111.1,2222.2,3333.3,3888.9,
 RO=1698,   T =  50*300,
 ALCTAB=1,    ALPHAT=1,   ALSTAB=1,    CE=.232,  GAMBAR=1.4,   GAMINF=1.4,
 MACHNO=20,     MWO2=32,     MWSTR=32,     PSTAGTB=1,
 PSEXP=-.17, RIFXP=.5,
 HFTAB=.8E+7,
 NHW=14, MHW=1,   TTABHW=277.8,555.6,833.3,1111.1,1388.9,1666.7,1944.4,2222.2,
 2500,2777.8,3055.6,3333.3,3611.1,3888.9,
 HWTAB=-23240,.258E+6,.5555E+6,.8717E+6,.12071E+7,.15457E+7,.1901E+7,.22779E+7,
 .27265E+7,.3217E+7,.3951E+7,.4788E+7,.5927E+7,.6973E+7,
 QCTAB=300,2*.1E+7,2*300,    TTABQC=0,1,50,51,80,     NQC=5,   MQC=1,
 DELTAO=5*.05,      DELTMIN=.1E-7,     L=5,    S=10,    XO=.024,
 R=5*.1E+51,   RS=5*1,   THETA=5*90,   ZS=0,.006,.012,.018,.024,
 PRAT=5*1,   QRAT=1,.95,.9,.85,.8,
 RNSI=1,      CORDSY=1,
 EPSONF=.98,    STFBOL=.56697E-7,
 DELTAU=.0078125,   DTMAX=.0625,   ENDTAU=80,   PRFREQ=10,
 ERRORT=.001, MAXITT=5,
 IPLOT=1, MDMAX=.025,   PTMAX=850, PTMIN=350, RSSMAX=.1,   ZSMAX=.05,
 NTP=3,1,5,10, PLTIME=0,20,40,60,80,     $
```

The sample output listing for a right-circular cylinder is given below:

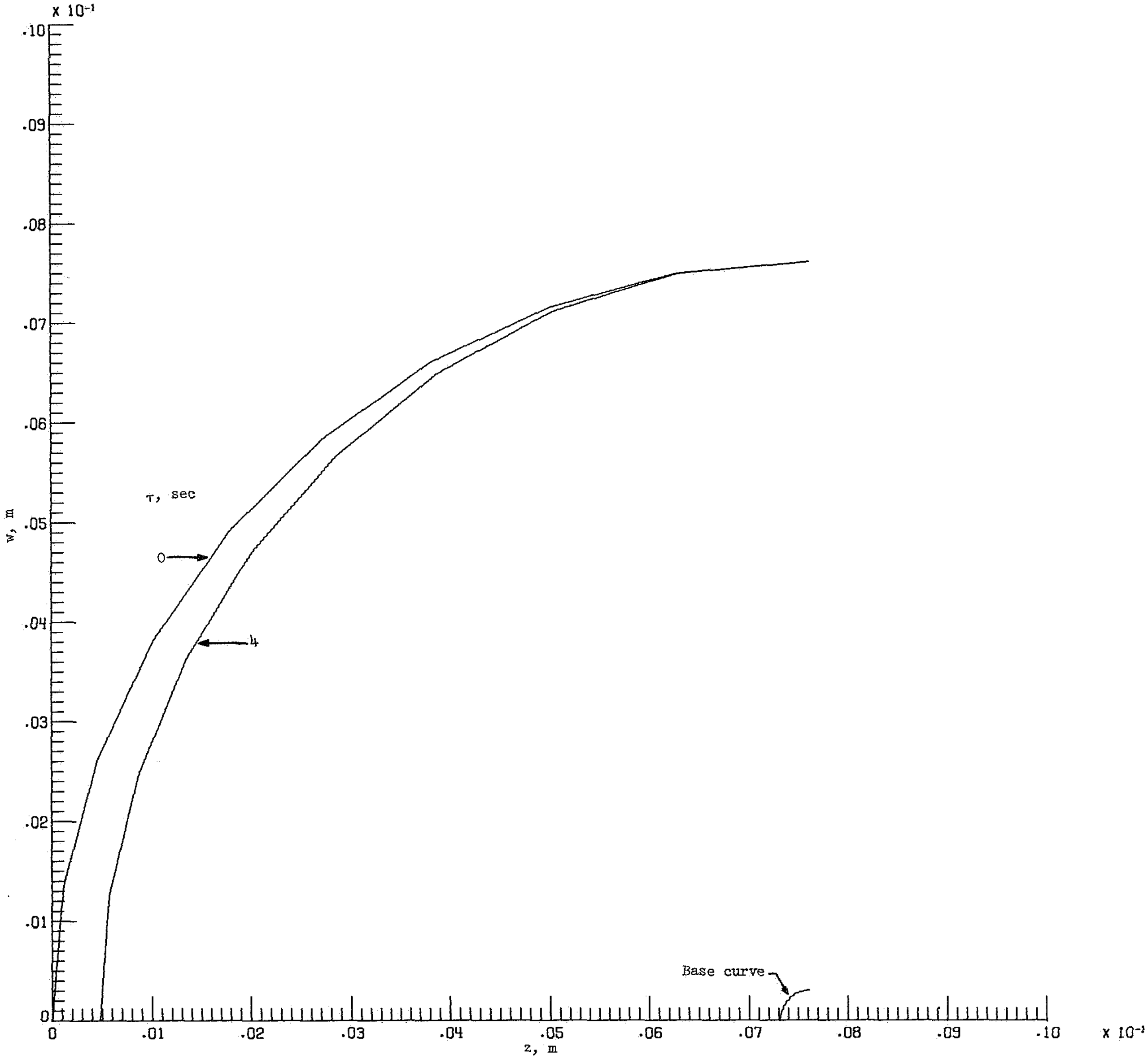
```
SAMPLE PROBLEM IN CARTESIAN COORDINATES


TAU=    10.0391              DELTAU=  .062500

               QC= 1.0000E+06     QR= 0.             HE= 8.0000E+06
               T(S,1)= 5.1910E+02

               TEMPERATURE (M,N)
ETA       X= 0.00000  .00600  .01200  .01800  .02400
1.000          519.9   517.9   514.5   511.0   509.0
 .889          478.9   478.1   476.5   474.9   474.2
 .778          447.6   447.3   446.6   445.8   445.5
 .667          423.2   423.1   422.8   422.4   422.3
 .556          404.2   404.2   404.0   403.8   403.8
 .444          389.6   389.6   389.5   389.5   389.4
 .333          378.8   378.8   378.7   378.7   378.7
 .222          371.3   371.3   371.3   371.3   371.2
 .111          366.9   366.9   366.9   366.9   366.9
0.000          365.4   365.4   365.4   365.4   365.4

               MDOT(N)--SURFACE MASS LOSS RATE
ETA       X=  0.00000      .00600      .01200      .01800      .02400
1.000         8.3173E-22  5.3341E-22  2.3829E-22  1.0543E-22  6.5989E-23

               MCDOT(N)--SURFACE MASS LOSS RATE DUE TO OXIDATION
1.000         0.          0.          0.          0.          0.

               DELTA(N)--MATERIAL THICKNESS
1.000         5.0000E-02  5.0000E-02  5.0000E-02  5.0000E-02  5.0000E-02

               QRAT(N)--RATIO OF LOCAL HEATING TO STAGNATION HEATING
1.000         1.0000E+00  9.5000E-01  9.0000E-01  8.5000E-01  8.0000E-01

               PRAT(N)--RATIO OF LOCAL PRESS TO STAG PRESS
1.000         1.0000E+00  1.0000E+00  1.0000E+00  1.0000E+00  1.0000E+00

               QS(N)--NET HEAT INPUT
1.000         9.7227E+05  9.2389E+05  8.7566E+05  8.2739E+05  7.7892E+05

               QRR(N)--RERADIATION
1.000         4.0590E+03  3.9987E+03  3.8923E+03  3.7882E+03  3.7299E+03

               QCOMB(N)--HEAT DUE TO COMBUSTION FOR OXIDATION
1.000         0.          0.          0.          0.          0.

      NO. ITER. COL.=   1     NO. ITER. ROW=   1     TOTAL NO. ITER.=     174     IROCOL=  1
```

## REFERENCES

1. Tompkins, Stephen S.; Moss, James N.; Pittman, Claud M.; and Howser, Lona M.: Numerical Analysis of the Transient Response of Ablating Axisymmetric Bodies Including the Effects of Shape Change. NASA TN D-6220, 1971.

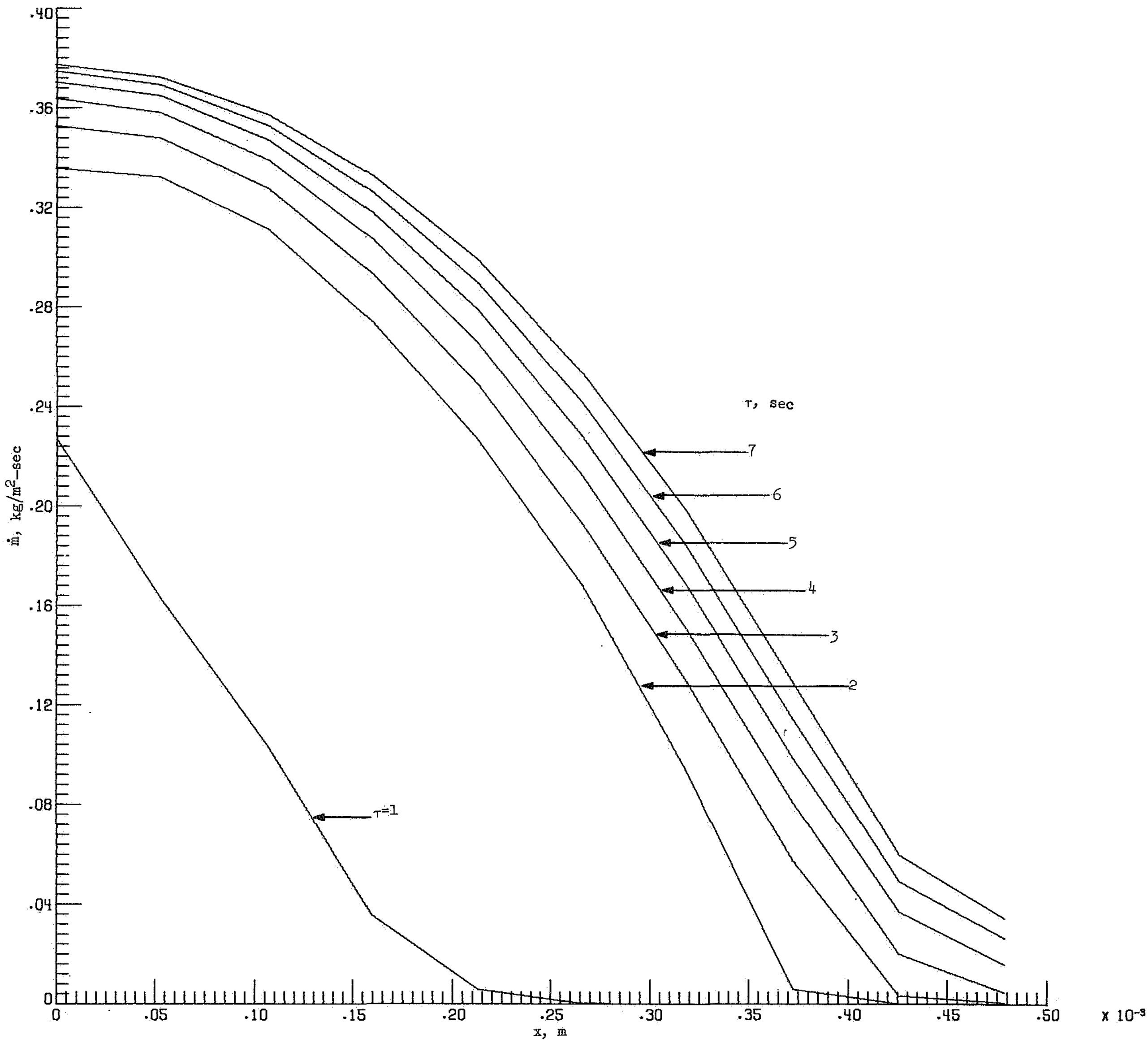
2. Gavril, Bruce D.; and Lane, Frank: Finite Difference Equation and Their Solution for the Transient Temperature Distribution in Composite, Anisotropic, Generalized Bodies of Revolution. Tech. Rep. No. 230 (Contract No. NOrd 18053), Gen. Appl. Sci. Lab., Inc., May 26, 1961.

3. Hovanessian, Shahen A.; and Pipes, Louis A.: Digital Computer Methods in Engineering. McGraw-Hill Book Co., Inc., c.1969.



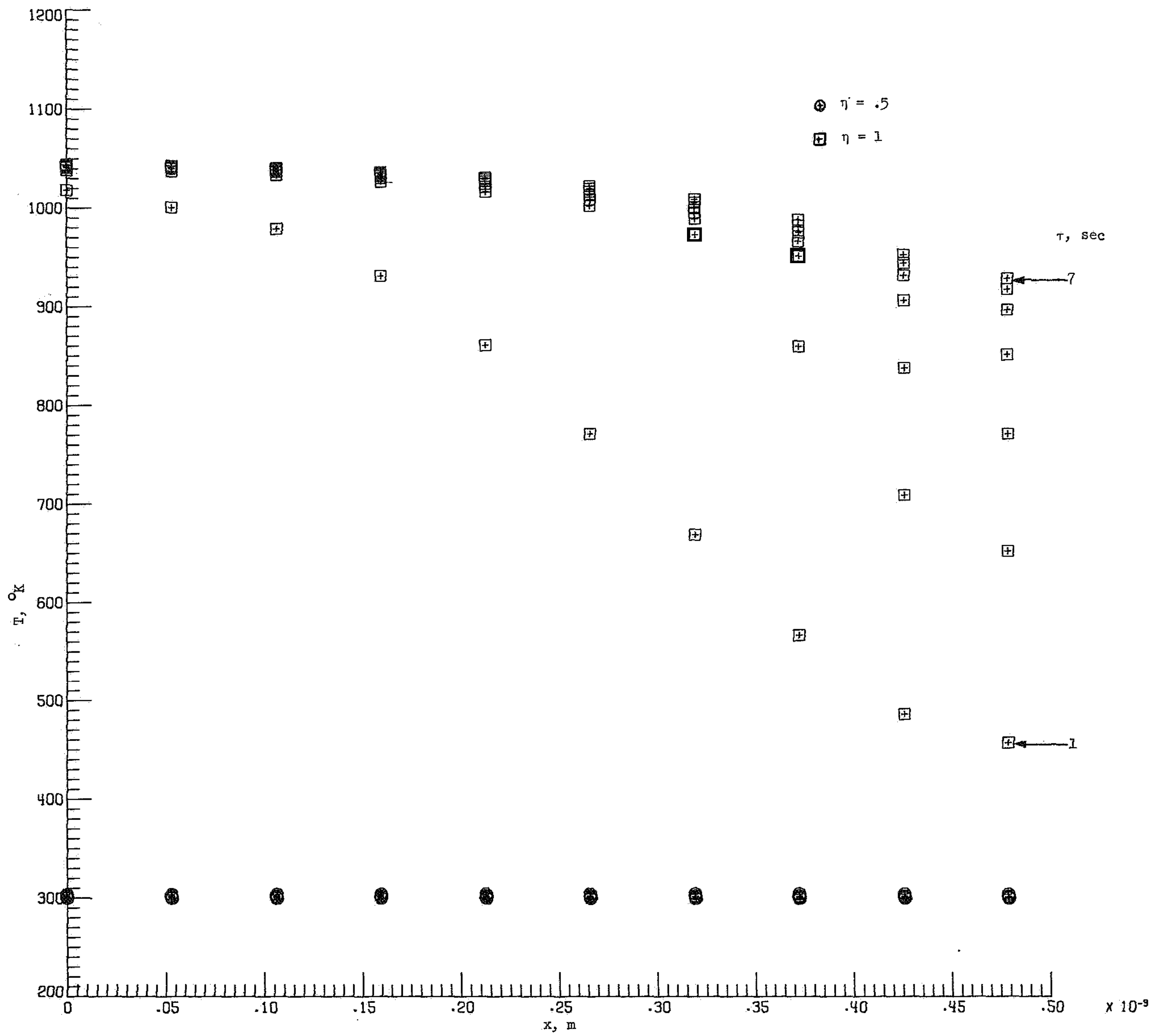

(a) Profile history.

Figure 1.- Computer-generated profile, mass loss, and temperature histories for a teflon hemisphere.



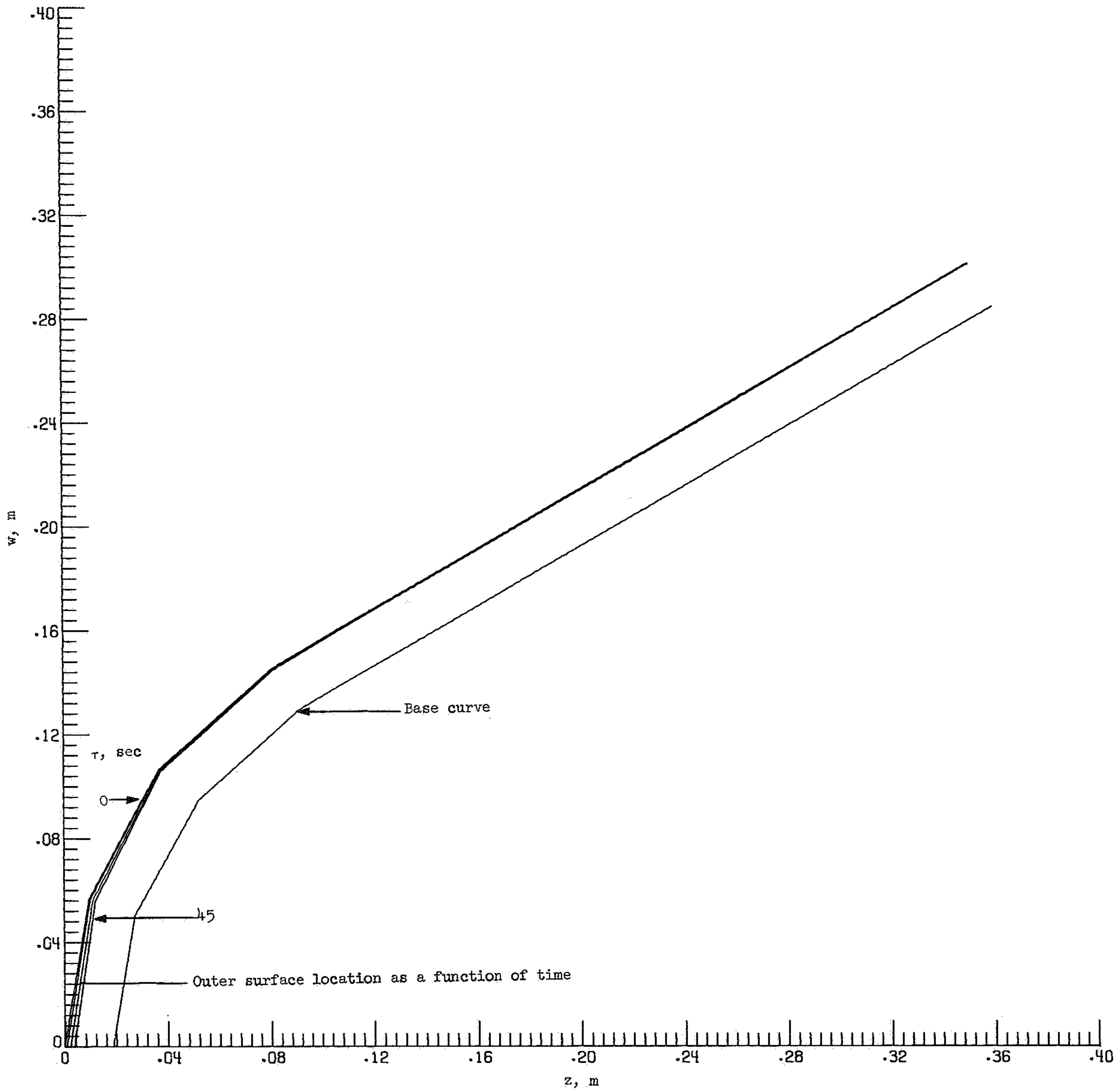

(b) Mass-loss-rate history.

Figure 1.- Continued.



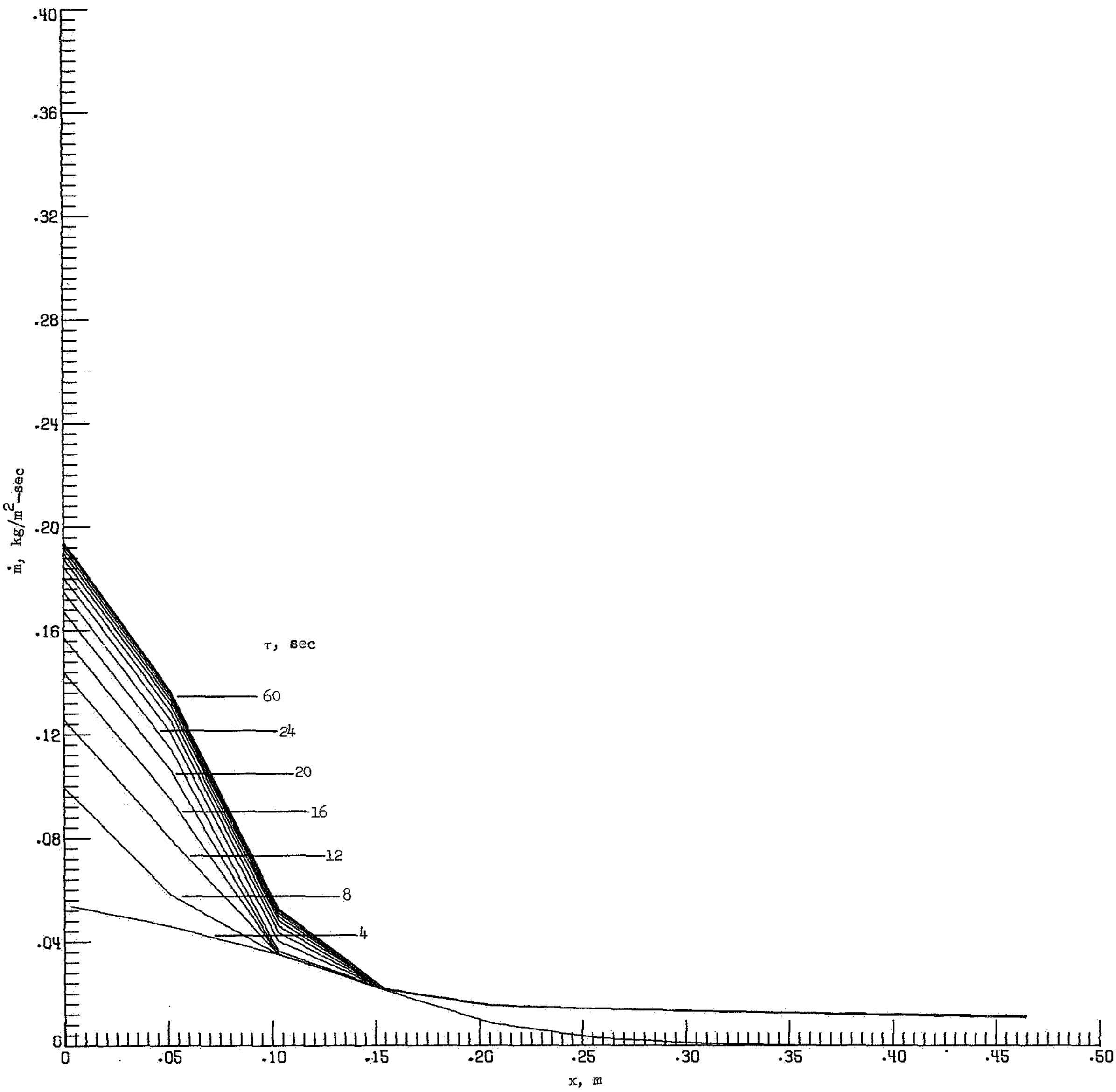

(c) Temperature history at times 1 to 7 sec in intervals of 1 sec at $\eta = 0.5$ and $\eta = 1$.

Figure 1.- Concluded.



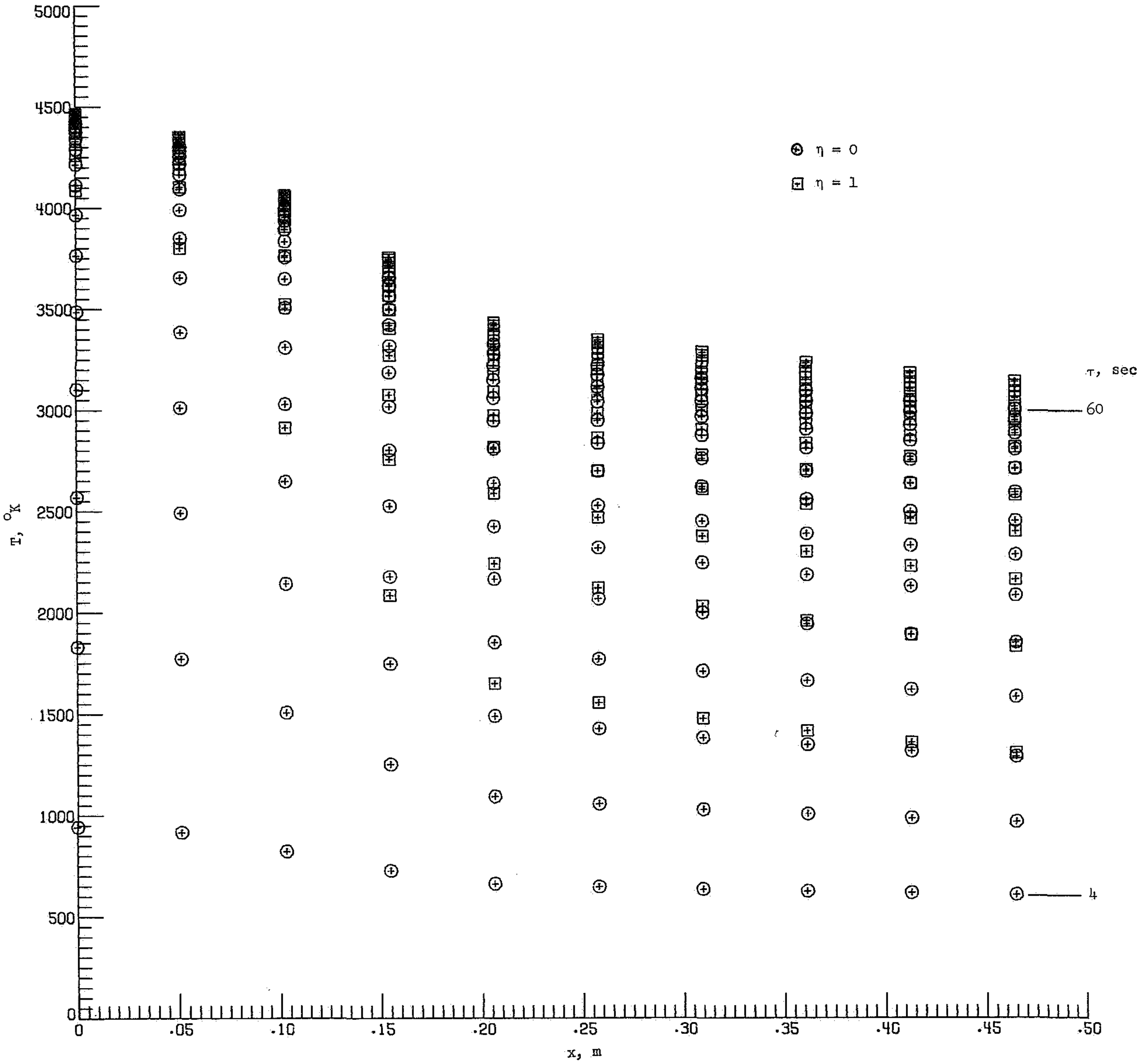

(a) Profile history at 15-sec intervals.

Figure 2.- Computer-generated profile, mass loss, and temperature histories for a graphite hemisphere-$30^{\circ}$ cone.



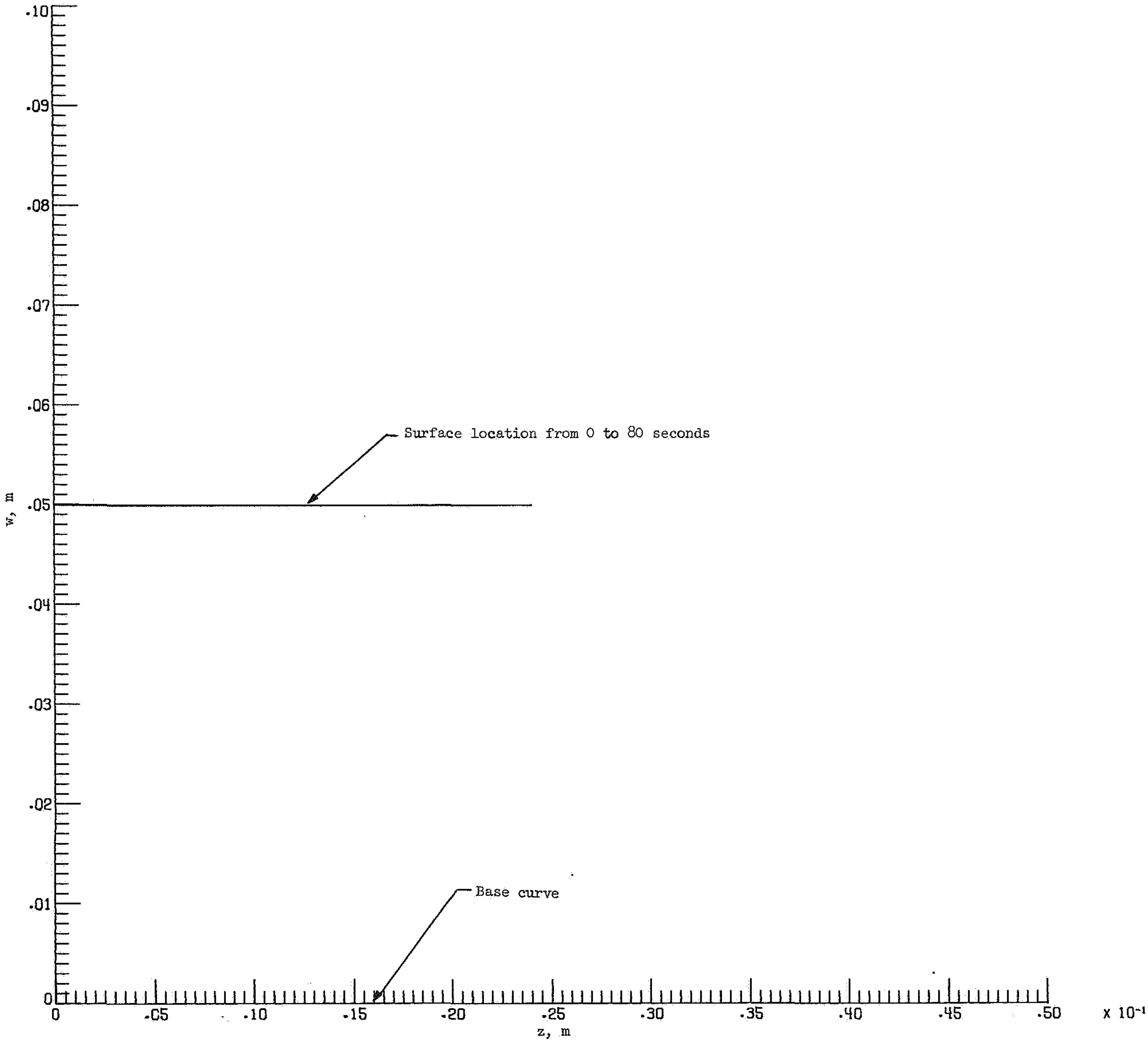
(b) Mass-loss-rate history at times 4 to 60 sec in intervals of 4 sec.

Figure 2.- Continued.



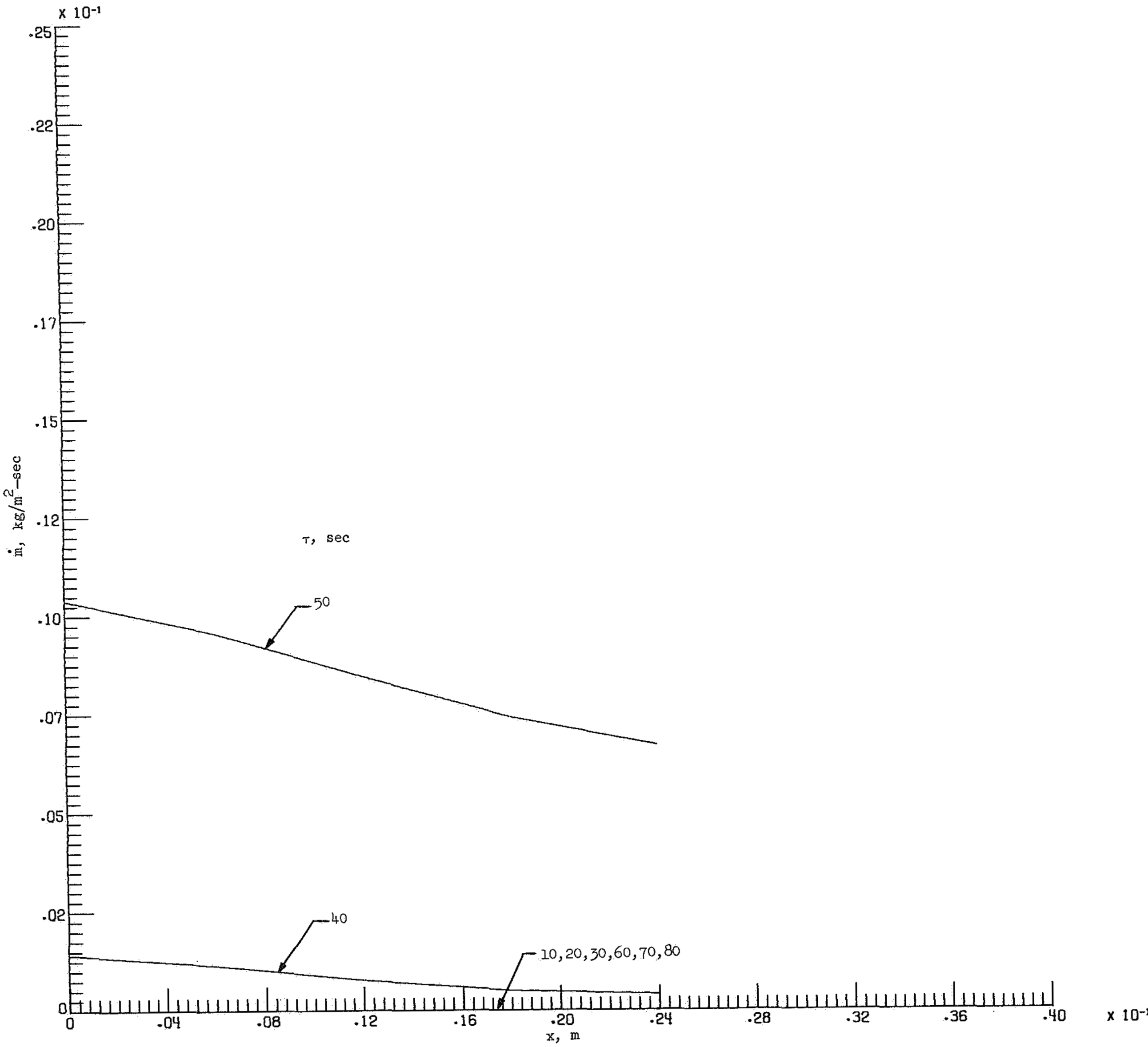

(c) Temperature history at times 4 to 60 sec in intervals of 4 sec at $\eta = 0$ and $\eta = 1$.

Figure 2.- Concluded.



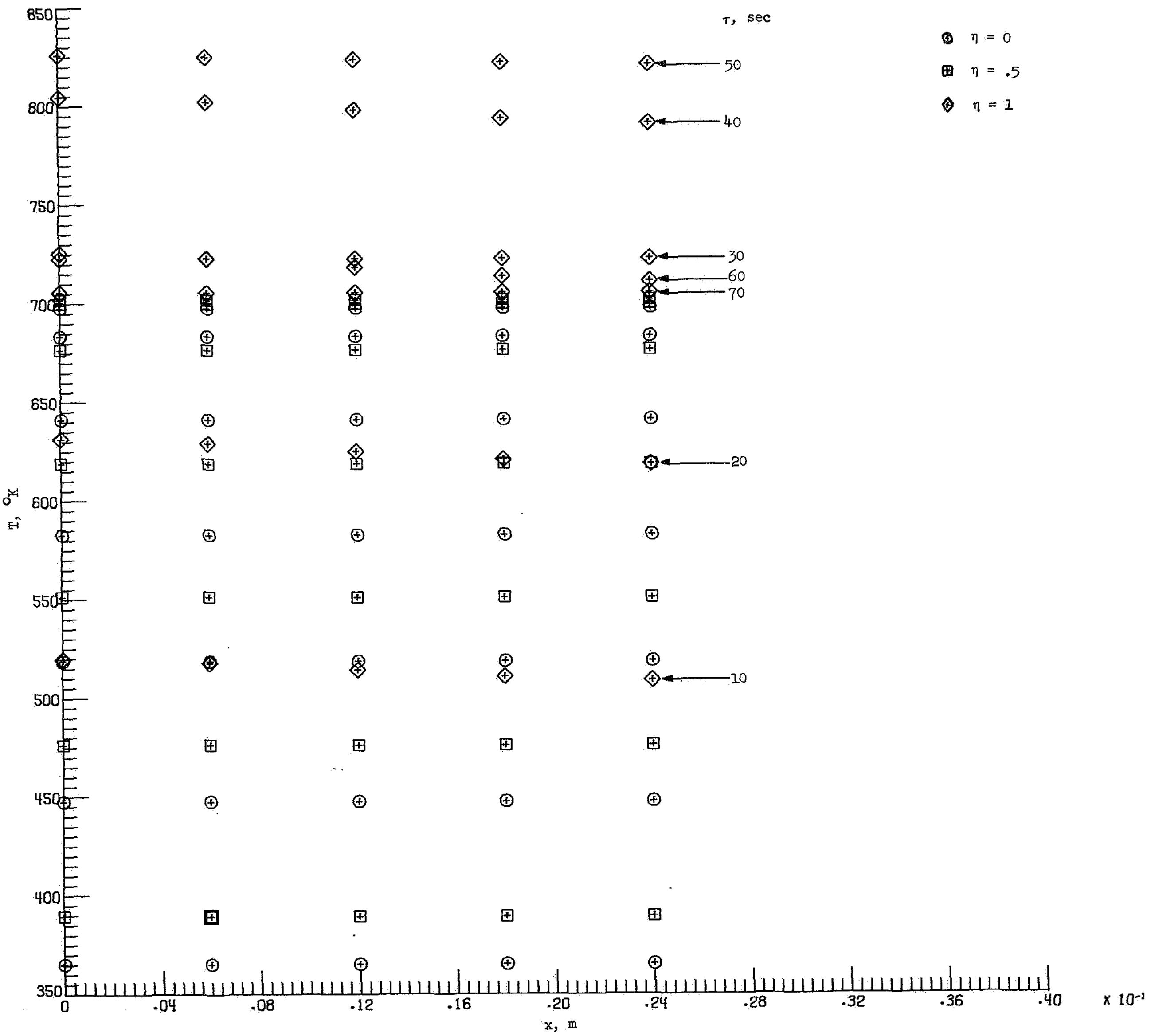

(a) Profile history.

Figure 3.- Computer-generated profile, mass loss, and temperature histories for a right-circular cylinder.




(b) Mass-loss-rate history.

Figure 3.- Continued.




(c) Temperature history at times 10 to 70 sec in intervals of 10 sec at $\eta = 0$, 0.5, and 1.

Figure 3.- Concluded.

*"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."*

—National Aeronautics and Space Act of 1958

# NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

*Details on the availability of these publications may be obtained from:*

**SCIENTIFIC AND TECHNICAL INFORMATION OFFICE**

**NATIONAL AERONAUTICS AND SPACE ADMINISTRATION**

**Washington, D.C. 20546**